IPSO FACTO

Issue #8
October, 1978

(A publication of the Association of Computer Experimenters)

TABLE OF CONTENTS

		PAGE
1.	ACE Executive & Meeting Schedule	2
2.	Editor's Remarks	3
3.	Mouse Trap Game	4
4.	Binary Quiz Program	4
5.	VIP Software Update	7
6.	State Display	8
7.	Software for CMOS 16X32 VIDEO SYSTEM	10
8.	The Three Keyboard Problem	14
9.	Programmer's Zodiac	15
10.	An RCA CDP 1802 System	19
11.	Software for 1K Video RAM	26
12.	Variation on a Theme	28
13.	Tiny Basic Square Root Routine Version 2	30
14.	Fast Memory Test	30
15.	An 1802 DMA Controller	31
16.	Program for Fast Memory Test	32
17.	Schematic for 1802 DMA Controller	34
18.	Memory Page Decoders	35
19.	Logic Probe	36
20.	Software for the "IPSO FACTO" Standard for KC Tapes	37
21.	A/D Notes	44
22.	Christmas Computer Music	45
23.	Hardware Basics	48
24.	Game of Life Update	52
25.	ACE Tutorial	54
26.	Diagrams for Game of Life Update	55
27.	Letters to the Editor	61
28.	Items for Sale	68
29.	Letters of Contact	69
30.	Minutes of ACE Meetings	70
31.	Application Form	73

Editor: Bernie Murphy
Invaluable Assistants: Wayne Bowdish, Tom Crawford, Ken Bieber, Diane York, and all contributors to this issue.

Information furnished by IPSO FACTO is believed to be accurate and reliable. However, no responsibility is assumed by IPSO FACTO or the Association of Computer Experimenters for its use; nor for any infringements of patents or other rights of third parties which may result from its use.

All Newsletter correspondence should be sent to:

Bernie Murphy
102 McCrany Street,
Oakville, Ontario
CANADA L6H 1H6

ACE EXECUTIVE COMMITTEE

In accordance with the Constitution, the 1978-79 Executive Committee approved at the Annual General Meeting is:

President	KEN BEVIS	220 Cherry Post Drive, Missisauga, Ontario, L5A 1H9 (277-2495)
Past President	TOM CRAWFORD	50 Brentwood Drive, Stoney Creek, Ontario, L8G 2W8 (662-5264)
Secretary/ Treasurer	GEORGE YORK	60 Chester Road, Stoney Creek, Ontario, L8E 1Y2 (664-5264)
Newsletter Editor	BERNIE MURPHY	102 McCrany Street, Oakville, Ontario, L6H 1H6 (845-1630)
Program Co-ordinator	BERT DEKAT	P.O. Box 137, Lynden, Ontario L0R 1T0 (647-3931)
Training Co-ordinator	ROD DORE	660 Oxford Road, Unit 32, Burlington, Ontario, L7N 3M1 (681-2456)
Hardware Co-ordinator	FRED FEAVER	105 Townsend Avenue, Burlington, Ontario, L7T 1Y8 (637-2513)
Membership Co-ordinator	WAYNE BOWDISH (Temporary)	149 East 33rd Street, Hamilton, Ontario, L8V 3T5 (388-7116)
Newsletter Publishing Committee	DENNIS MILDON	44 Wildewood Avenue, Hamilton, Ontario, L8T 1X3 (385-0798)
	JOHN HANSON	955 Harvey Place, Burlington, Ontario, L7T 3E9 (637-1076)

There are still some positions to be filled eg. software and membership co-ordinators. Any volunteers for these positions or to help with any other activity should contact Ken Bevis.

CLUB MEETINGS

Unless notice to the contrary, the following is the meeting schedule for the remainder of the year, also, all meetings will be held in the Stelco Wilcox St. Auditorium.

DATE	TUTORIAL/MEETING	TUTORIAL
Nov 14	7:00/8:00	
Nov 28		7:00
Dec 12	7:00/8:00	

BACK ISSUE REPRINT POLICY

The cost of reprints of back issues will be:

1 complete set of issues #1 to 6 ..... $12.00
Per copy price of single issue ..... $ 2.50

The last year has been quite an exciting year for many 1802 users. Many of us have progressed from 256 byte one card systems to full blown systems running BASIC, ASSEMBLER, and sophisticated machine language programs. Some 1802 owners are contempleting interfacing disc drives and there is even talk of implementing software such as PASCAL and FORTH for the 1802.

If you are thinking of working on PASCAL or FORTH for the 1802, please write up your ideas in the form of an article. Let us not have any of us re-inventing the wheel!

HARDWARE PROJECTS

With the advent of the 64K bit single voltage source dynamic RAM, the price of 8K and 16K dynamic memory chips should fall drastically in the next year or so. Do you remember when the 2102 RAM was a $5.00 part?

Has anyone interfaced any dynamic memory to the 1802? With the price of an 8K chip in the $5.00 range, the hassle of using dynamic memory becomes worthwhile (8K for $40.00!).

What about S-100 memory cards? If you have got some of these gems going, please tell all!

NEW 1800 PRODUCTS FROM RCA

I understand that the new RCA microprocessor is called an 1804. Write your friendly RCA distributor for the latest product guide. The part number of the new guide is MPG-180B.

SOFTWARE PROJECTS

Many Radio Amateurs have a CW transit program going. How about that elusive CW read program? Someone has done it for the 6502--how about the 1802?

GETTING YOURSELF UP TO SPEED

If you have been missing out on all the fun because of lack of knowledge, the time to upgrade is NOW. You know what happened in the 1960's to all the Engineers and Technicians who said that transistors were "gimmicks" and that tubes were here to stay?--they are now unemployed! The articles by Peter A. Stark in Kilobaud magazine are excellent if your digital hardware knowledge is a little thin or non-existant.

The current software situation for the 1802 leaves much to be desired. I believe Osborne has a book on 1802 programming. A book review would be welcomed.

BOOTLEG SOFTWARE

Quite a few of us have TINY BASIC running on the 1802. If you have a "borrowed" copy, please send your money to the author. By using a bootleg copy, you are denying the author his much deserved royalty. In my opinion, TINY BASIC by Itty Bitty Computers Inc. at $5.00 is the software bargain of the century. What has been accomplished in 2K bytes of code is truly amazing.

I understand there exists a 5K version of BASIC for the 1802--is anybody running it?

My space allotment is up; as one of our readers suggested--take an ELF II lunch...

MOUSE TRAP GAME

Jim LaVeck
Route 1, Box 150
Dexter, New York 13634

Dear Tom,

Enclosed is my Poor Man's Mousetrap game, for those of us without a video display. When run, the program displays 0-F sequentially. When 8 is displayed, you must push the input switch. After 10 attempts at catching the mouse, your score is displayed, and the Q light is turned on.

POOR MAN'S MOUSE TRAP GAME

```
00   F8  00  B1  A5    Zero Turns
04   F8  32  A1        Load Address of Score
07   F8  0A  A4        Load # of Turns
0A   E1  64  21        Display Count
0D   F0  FC  01  51    Increment Count
11   F8  0F  F2  51    AND Count with 0F
15   F8  15  B2        Load Delay of Count
18   22                Decrement Delay
19   37  20            Wait for Input =1
1B   92                Get R(2).1
1C   3A  18            GOTO Decrement if delay Not Zero
1E   30  0A            If Delay Zero, Display Count
20   37  20            Debounce Input
22   F8  09  F5        D - 9
25   24                Decrement # Turns.
26   3A  29            GOTO Decrement of Turns, if a miss
28   15                Increment Score
29   84                Get # Turns
2A   3A  0A            Display If Turns = Zero
2C   85                R(5).0→D
2D   51  64            Display Score
2F   7B                Set Q
30   00                Stop
```

BINARY QUIZ PROGRAM

Mike Cohen
2255 Barker Ave.
Bronx, N.Y. 10467

Dear Tom,

Enclosed is a Binary Quiz program that will run on any ELF with PIXIE graphics and a HEX keyboard connected to INP 4. When the program is started, the display area is cleared and pressing the input switch(EF4) will generate and display a random number between 0 and 255. Pressing the input switch will enter your answer. If the answer is correct, the LED display(OUT 4) will show "CC" and Q will be turned on; if wrong, Q will be turned off and the Display will show "FF". Releasing the switch will show a new number. There is no time limit.

BINARY QUIZ PROGRAM (CONT'D)

ADDR	CODE	STATEMENT	COMMENT
00	90	GHI R0	
01	B1	PHI R1	Clear Register 1.1
02	B2	PHI R2	Clear Register 2.1
03	B3	PHI R3	Clear Register 3.1
04	B8	PHI R8	Clear Register 8.1
05	F8 31	LDI #31	
06	A3	PLO R3	Main Program Counter
08	F8 12	LDI #12	
0A	A2	PLO R2	Stack Pointer
0B	F8 15	LDI #15	
0D	A1	PLO R1	Interrupt Program Counter
0E	D3	SEP R3	Go To Main Program
0F	00 00		STACK
11	00 00		
13	72	LDXA	Restore Contents of D
14	70	RET	Return from Interrupt
15	22	DEC R2 (INTERRUPT)	
16	78	SAV	Save X,P
17	22	DEC R2	
18	52	STR 2	Save Contents of D
19	C4 C4 C4	NOP	Delay
1C	F8 00	LDI #00	
1E	B0	PHI R0	
1F	F8 00	LDI #00	
21	A0	PLO R0	Set DMA Pointer
22	80	GLO R0	
23	E2	SEX R2	
24	E2 20 A0		Refresh Display (8 DMA Cycles)
27	E2 20 A0		"
2A	E2 20 A0		"
2D	3C 22	BN1 22	Repeat 32 Times
2F	30 13	BR 13	Return
31	E2	SEX R2	
32	69	INP 1	Turn on Video Display
33	30 84	BR 84	Clear Screen
35	1E	INC RE	Increase Randowm Number Counter
36	37 35	B4 35	Repeat Until Input Switch is Released
38	8E	GLO E	Get Random Number...
39	AF	PLO RF	..and Store in RF.0
3A	F8 08	LDI #08	Set Bit Counter
3C	A5	PLO 5	
3D	F8 78	LDI #78	Table of Bytes for Zero...
3F	A6	PLO R6	..to Register 6.0
40	F8 7E	LDI #7E	Table of Bytes for One...
42	B6	PHI R6	..to Register 6.0
43	F8 B8	LDI #B8	Display Location...
45	A8	PLO R8	..to Register 8.0
46	A9	PLO R9	.. and 9.0
47	8F	GLO F	Get Random Byte...
48	FE	SHL	..Shift Left to Test Bit 7
49	AF	PLO F	..and Put It Back For Storage

BINARY QUIZ PROGRAM (CONT'D)

ADDR	CODE	STATEMENT	COMMENT
4A	33 50	BDF 50	Test Bit for a One
4C	86	GLO 6	Get Table for Zero...
4D	A7	PLO 7	..Put in R7
4E	30 52	BR 52	Skip Next Step
50	96	GHI 6	Get Table for One
51	A7	PLO 7	put in R7
52	47	LDA 7	Get Byte from Table
53	32 5C	BZ 5C	Test for End of Table
55	58	STR 8	Store in Display Location
56	88	GLO 8	
57	FC 08	ADI #08	Advance to Next Line
59	A8	PLO 8	
5A	30 52	BR 52	Repeat Until End of Table
5C	25	DEC R5	Decrease Bit Counter
5D	85	GLO R5	
5E	32 65	BZ 65	Test For End of Byte
60	19	INC R9	
61	89	GLO R9	Advance to Next Bit Position
62	A8	PLO R8	
63	30 47	BR 47	Repeat For All 8 Bits
65	3F 65	BN4 65	Wait for Input Switch to be Pressed
67	E2	SEX 2	
68	6C	INP 4	Get Answer
69	8E	GLO RE	Get Display Byte
6A	F3	XOR	Test Answer
6B	E3	SEX 3	
6C	32 73	BZ 73	Branch if Correct
6E	64 FF	OUT 4	Display "FF"
70	7A	REQ	And Turn Q-light Off
71	30 35	BR 35	Generate New Random Number
73	64 CC	OUT 4	Display "CC"
75	7B	SEQ	And Turn Q-light On
76	30 35	BR 35	
78	F0 90 90	90 F0 00	Table for Zero
7E	60 20 20	20 70 00	Table for One
84	F8 E7	LDI #E7	Top of Area to be Cleared
86	A4	PLO R4	
87	E4	SEX 4	
88	F8 00	LDI #00	Clear Stack Location
8A	73	STXD	& Advance Location
8B	84	GLO 4	
8C	FB AF	XRI #AF	Test for End of Stack
8E	00	WAIT	Delay for One Display Frame
8F	3A 88	BNZ 88	Repeat to End of Stack
91	3F 91	BN4 91	Wait for Input Switch to be Pressed
93	30 35	BR 35	Go To 35 (Generate Random Number)

VIP SOFTWARE UPDATE

Richard Blessing
Apt 1, 237 Baywood Dr.
Pass Christian, MS 39571

I have enclosed an update to my original artical (Issue #7, page 16) and the data set to go along with the SNOOPY picture.

RCA has advised me that you should not apply -5 volts to the N2 line from the 1802 as I have in my design. I have had no problems but I did change my set up so the reset signal comes from the Q line. This also isolates N2. See Figure 1.

```
?MD00 1B0        DATA SET FOR SNOOPY
0D00 0000 0000 0000 0000 0000 0F01 C000 0000;
0D10 0000 1801 0000 0000 0000 FFE2 0000 0000;
0D20 0000 3004 0000 0000 0007 C03C 0000 0000;
0D30 001C 000E 0000 0000 0070 0013 8000 0000;
0D40 01C0 0021 C000 0000 0300 0040 E000 0000;
0D50 0600 0000 3C00 FE00 0700 0000 3F1F 8F80;
0D60 0E00 0000 E3FC 01E3 0E00 0007 8000 0037;
0D70 1C00 000E 0000 001F 1FE0 0000 0000 000F;
0D80 3FC0 0000 0000 0002 3FB8 0000 0000 0003;
0D90 7EB8 0000 0000 0001 77F8 0000 0000 0003;
0DA0 F6EC 0000 0600 0006 F7EC 0000 0F00 000C;
0DB0 EFFC 0000 0600 000C E6F6 0000 0080 0030;
0DC0 EFF6 0000 00C0 0010 FFFB 0000 0060 00E0;
0DD0 FFFB 0000 0030 0300 FFDD 8000 003F 9E00;
0DE0 FFF9 8000 0030 F000 FFFB 8000 00C0 0000;
0DF0 7FFA E000 0380 0000 7FFE F000 0600 0000;
0E00 37FC CC00 1800 0000 37FE C700 1800 0000;
0E10 13FF C700 3000 0000 1FFF C180 3000 0000;
0E20 0FFF C0C0 3000 0000 07FF C060 3000 0000;
0E30 03FE 8018 0C00 0000 01FC 8018 0600 0000;
0E40 007F 0018 0F80 0000 0000 0008 7F00 0000;
0E50 0000 000F E1C0 0000 0000 001F 01C0 0000;
0E60 0000 001C 00E0 0000 0000 000C 00E0 0000;
0E70 0000 000C 0070 0000 0000 000D 0038 0000;
0E80 0000 000D 0418 0000 0800 000D 0438 0000;
0E90 1000 000D 040E 0000 2000 0009 040C 0000;
0EA0 2000 000A 0806 0000 2000 000A 0803 0000
```

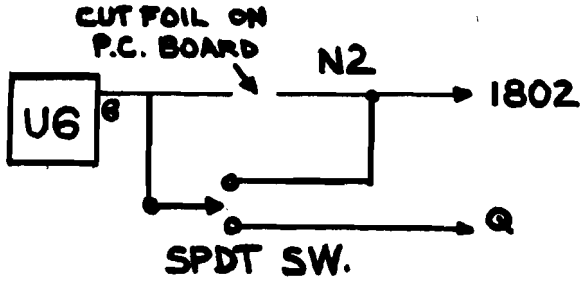


FIG.1

ADD A STATE DISPLAY TO THE COSMAC ELF

David Grenewetzki
23145 Bigler St.
Woodland Hills, CA 91364

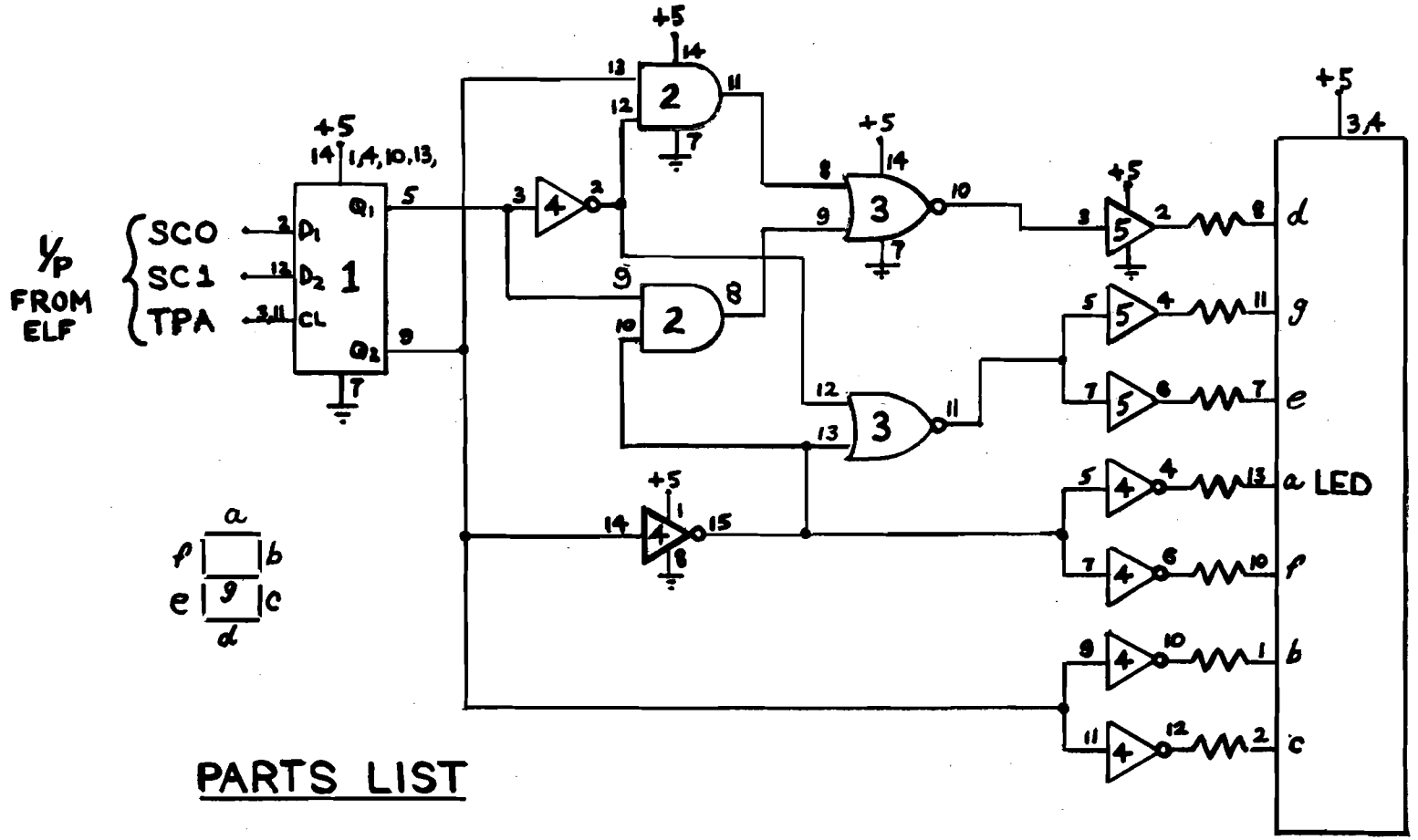
If you've added single-step operation to your COSMAC ELF microprocessor*, the following circuit will make the task of debugging programs quite a bit easier. The circuit in Figure 1 monitors the state code lines from the ELF and displays a letter corresponding to the current machine state. If the machine is in a "fetch" operation, an "F" is shown on the display. Similarly, a "D" for DMA, an "I" for interrupt and an "E" for execute are displayed.

STATE CODES SC0	SC1	STATE	DISPLAY
1	1	Interrupt	I
1	0	Execute	E
0	0	Fetch	F
0	1	DMA	d

In operation the circuit samples the ELF's state code lines, SC0 and SC1, during the TPA timing pulse and decodes the lines to drive the segments of the LED display. Construction of the circuit is non-critical and almost any wiring technique may be used. Since CMOS circuits are used, the SC0, SC1 and TPA lines may be directly connected to the ELF without the use of buffer circuitry. The circuit uses a maximum current of about 50 mA with most of the power going to the LED.

*One simple method for implementing a single-step mode in the 1802 microprocessor is shown in RCA publication MPM-201A, "User Manual for the CDP1802 Microprocessor."

FIGURE 1: STATE DISPLAY CIRCUIT FOR COSMAC ELF



PARTS LIST

IC 1	74C74	DUAL D FLIP FLOP
IC 2	74C08	QUAD 2 INPUT AND
IC 3	4001	QUAD 2 INPUT NOR
IC 4	4049	INV. HEX BUFFER
IC 5	4050	NON-INV. HEX BUFFER

R1-R7 470 Ω 1/4 W

LED 7 SEGMENT COM. ANODE LED DISPLAY (RADIO SHACK 276-053 OR EQUIV.)

SOFTWARE FOR THE CMOS 16X32 VIDEO SYSTEM

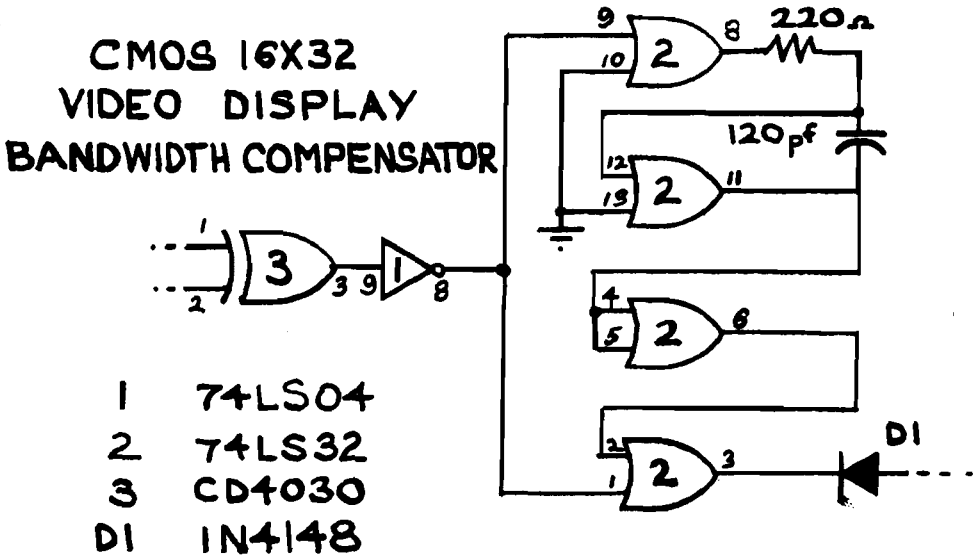
G.E. Millar
P.O. Box 1412
Parksville B.C. VOR 2SO

I am enclosing a cursor control program which I have written for use with my video display (IPSO FACTO issue #6, Page 23) which will handle the following commands.

1. 01 - home cursor
2. 08 - back space
3. 09 - cursor right
4. 0A - line feed
5. 0B - cursor up
6. 0D - carriage return
7. 11 - scroll up
8. 12 - errase from cursor to end of screen
9. 18 - clear screen
10. 20→FF - display ASCII character

(all inputs from 00→19 not specified above are ignored). The program can reside in any page of memory, I am using 8300. My display memory is located at 8E00→8FFF. If the cursor increments beyond the end of screen while entering data, it automatically scrolls up. The first or top line of data is scrolled off the screen and lost. My ASCII keyboard input entry is at KBIN: and data is passed in RF.1 which is saved on stack and restored at exit of video subroutine. Dispaly entry is at VIDEO: and data is passed in RF.1. Register RC is used as cursor pointer and all other registers used are saved on entry and restored on exit. The cursor up will roll under if cursor leaves top of screen and the back space will enter bottom right of screen if decremented past the home position.

I have made one modification to my video display. I am using an <u>old</u> tube type G.E. 11" TV, which I am feeding at the antenna terminals through a video cube. The T.V. has a narrow bandwidth causing the vertical portions of all characters to be lighter than the horizontal portions. To overcome this problem I have installed the enclosed circuit (Bandwidth Compensator used in the TVT6) between U1 pin 8 and D1. The 74LS32 occupies the unused socket U5. This improves the clarity of the characters tremendously.



```
                        ;
                        ;  INPUT ENTRY AT "KBIN" ... OUTPUT ENTRY AT "VIDEO"
                        ;  VIDEO DISPLAY LOCATION #8E00 TO #8FFF
                        ;                ( R12 = CURSOR )
                        ;
        =  83 00        .ORG    #8300   ; STARTING ADDRESS OF ROUTINE

8300  60        VEXIT:  IRX
8301  72                LDXA
8302  AE                PLO     R14
8303  72                LDXA
8304  BE                PHI     R14     ; RESTORE R14
8305  72                LDXA
8306  AF                PLO     R15
8307  F0                LDX
8308  BF                PHI     R15     ; RESTORE R15
8309  D5                +RETRN
                        ;
830A  3C 0A     KBIN:   BN1     KBIN    ; WAIT FOR KB STROBE
830C  69                INP     1       ; INPUT ASCII CHARACTER
830D  BF                PHI     R15     ; SAVE IT
                        ;
830E  9F 73 8F  VIDEO:  +PUSHR  R15     ; SAVE R15
8311  73
8312  9E 73 8E          +PUSHR  R14     ; SAVE R14
8315  73
8316  9C                GHI     R12
8317  FB 8E             XRI     #8E     ; IS CURSOR ON SCREEN
8319  32 24             BZ      ONPAGE
831B  F6                SHR
831C  32 24             BZ      ONPAGE  ; YES
831E  F8 00 AC          +DLDI   #8E00,R12       ; NO - HOME CURSOR
8321  F8 8E BC
8324  0C        ONPAGE: LDN     R12
8325  FA 7F             ANI     #7F     ; REMOVE CURSOR
8327  5C                STR     R12
8328  9F                GHI     R15
8329  FD 1F             SDI     #1F     ; BRANCH IF CHARACTER
832B  3B 56             BNF     CHAR
832D  9F                GHI     R15     ; MACHINE OP.
832E  FB 0A             XRI     #0A
8330  32 6F             BZ      LF.OP   ; "LF" ( 04 )
8332  FB 07             XRI     #07
8334  32 61             BZ      CR.OP   ; "CR" ( 0D )
8336  FB 0C             XRI     #0C
8338  32 67             BZ      HOMEC   ; "HOME CURSOR" ( 01 )
833A  9F                GHI     R15
833B  FB 08             XRI     #08
833D  32 B2             BZ      LEFTC   ; "BACK SPACE" ( 08 )
833F  FB 01             XRI     #01
8341  32 58             BZ      RIGHTC  ; "CURSOR RIGHT" (HT) ( 09 )
8343  FB 02             XRI     #02
8345  32 A7             BZ      UP.CUR  ; "CURSOR UP" (VT) ( 0B )
8347  9F                GHI     R15
```

```
 1  8348 FB 18            XRI     #18
 2  834A 32 BD            BZ      CLEARC  ; "CLEAR SCREEN" (CAN) ( 18 )
 3  834C FB 09            XRI     #09
 4  834E 32 7E            BZ      SCROLL  ; "SCROLL UP" ( 11 )
 5  8350 FB 02            XRI     #02
 6  8352 32 C3            BZ      ERRACE  ; "ERASE TO END OF SCREEN" ( 12
 7  8354 30 5B            BR      ADDCUR
 8                        ;
 9  8356 9F       CHAR:   GHI     R15
10  8357 5C               STR     R12     ; STORE CHARACTER
11  8358 1C       RIGHTC: INC     R12     ; INC CURSOR
12  8359 30 79            BR      CKPAGE  ; CHECK FOR END OF PAGE
13  835B 0C       ADDCUR: LDN     R12
14  835C FC 80            ADI     #80
15  835E 5C               STR     R12     ; ADD CURSOR
16  835F 30 00            BR      VEXIT   ; RETURN
17                        ;
18  8361 8C       CR.OP:  GLO     R12     ;"CR"
19  8362 FA E0            ANI     #E0
20  8364 AC               PLO     R12
21  8365 30 5B            BR      ADDCUR
22                        ;
23  8367 F8 00 AC HOMEC:  +DLDI   #8E00,R12       ; "HOME CURSOR"
24  836A F8 8E BC
25  836D 30 5B            BR      ADDCUR
26                        ;
27  836F 8C       LF.OP:  GLO     R12     ; "LF"
28  8370 FC 20            ADI     #20
29  8372 AC               PLO     R12
30  8373 3B 79            BNF     CKPAGE  ; CHANGE PAGE ?
31  8375 9C               GHI     R12     ;  YES
32  8376 FC 01            ADI     #01
33  8378 BC               PHI     R12
34  8379 9C       CKPAGE: GHI     R12
35  837A FB 90            XRI     #90     ; CURSOR OFF SCREEN ?
36  837C 3A 5B            BNZ     ADDCUR  ;  NO
37                        ;
38  837E 9C       SCROLL: GHI     R12
39  837F BE               PHI     R14
40  8380 8C               GLO     R12     ; STORE CURSOR LOCATION 1 LINE UP
41  8381 FF 20            SMI     #20
42  8383 AE               PLO     R14
43  8384 33 91            BDF     SCR1
44  8386 9C               GHI     R12
45  8387 FF 01            SMI     #01
46  8389 BC               PHI     R12     ; IS CURSOR OFF SCREEN ?
47  838A FB 8E            XRI     #8E
48  838C 32 91            BZ      SCR1    ;  NO
49  838E F8 8F            LDI     #8F     ;  YES - ROLL UNDER
50  8390 BE               PHI     R14
51  8391 F8 00 AC SCR1:   +DLDI   #8E00,R12  ; HOME CURSOR
52  8394 F8 8E BC
53  8397 8C               GLO     R12
54  8398 FC 20            ADI     #20
```

```
 1  839A  AF          PLO   R15
 2  839B  9C          GHI   R12
 3  839C  BF          PHI   R15
 4  839D  4F  SCR2:   LDA   R15     ; GET DATA
 5  839E  5C          STR   R12     ;  STORE IT
 6  839F  1C          INC   R12     ;  NEXT
 7  83A0  9F          GHI   R15
 8  83A1  FB 90       XRI   #90     ; FINISHED ?
 9  83A3  3A 9D       BNZ   SCR2    ;  NO
10  83A5  30 C7       BR    CLR1    ;  YES - CLEAR LAST LINE
11                    ;
12  83A7  8C  UP.CUR: GLO   R12
13  83A8  FF 20       SMI   #20     ; MOVE CURSOR 1 LINE UP
14  83AA  AC          PLO   R12     ; CHANGE PAGE ?
15  83AB  33 B3       BDF   LTC     ;  NO
16  83AD  9C          GHI   R12     ;  YES
17  83AE  FF 01       SMI   #01
18  83B0  BC          PHI   R12
19  83B1  38          SKP
20  83B2  2C  LEFTC:  DEC   R12     ; BACK SPACE
21  83B3  9C  LTC:    GHI   R12
22  83B4  FB 8D       XRI   #8D     ; OFF PAGE ?
23  83B6  3A 5B       BNZ   ADDCUR  ;  NO
24  83B8  F8 8F       LDI   #8F     ;  YES - ROLL UNDER
25  83BA  BC          PHI   R12
26  83BB  30 5B       BR    ADDCUR
27                    ;
28  83BD  F8 00 AC CLEARC: +DLDI  #8E00,R12  ; "CLEAR SCREEN"
29  83C0  F8 8E BC
30  83C3  9C  ERRACE: GHI   R12     ; "CLEAR TO END OF SCREEN"
31  83C4  BE          PHI   R14
32  83C5  8C          GLO   R12
33  83C6  AE          PLO   R14
34  83C7  F8 20 CLR1: LDI   #20     ; LOAD "SPACE"
35  83C9  5C          STR   R12     ; STORE IT
36  83CA  1C          INC   R12     ; NEXT
37  83CB  8C          GLO   R12     ;  DONE ? (PAGE)
38  83CC  3A C7       BNZ   CLR1    ;   NO
39  83CE  9C          GHI   R12     ;   YES
40  83CF  FB 90       XRI   #90     ; END OF SCREEN ?
41  83D1  3A C7       BNZ   CLR1    ;  NO
42  83D3  9E          GHI   R14     ;  YES - RESTORE CURSOR LOCATION
43  83D4  BC          PHI   R12
44  83D5  8E          GLO   R14
45  83D6  AC          PLO   R12
46  83D7  30 5B       BR    ADDCUR
47                    .END
```

THE THREE KEYBOARD PROBLEM

Jim R. Smith
4629 North Shore Drive
Wichita Falls, TX, 76310

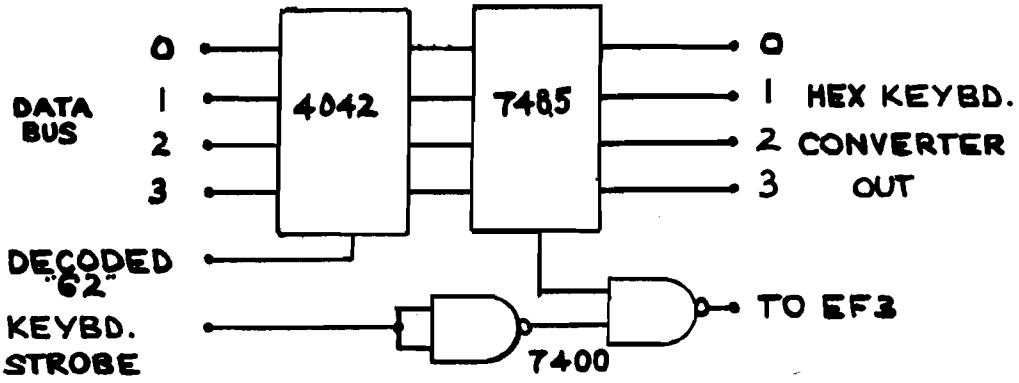
Once upon a time, I had three keyboards...an ASCII, an ELF II (74C922 and 74C173) and a keyboard using a 4515 for VIP programs.

I used the circuit in a recent BYTE magazine using a 7486 and a 7400 to convert from ASCII to hex. Then I had to pack the hex numbers. I used about 5 (actually 4½) 74LS74's (you use what you have) to accomplish this. Well, this took care of TINY and input port #4 (6C) which I use to direct my monitor.

Now I had only two keyboards. I had whispered all kinds of things to my 1802 trying to get the VIP system to work with keyboard inputs into #4 input port. I could get the operating system to understand me, but CHIP 8 remained deaf. That is why I had built the 4515 keyboard in the first place.

Until the light in the upstairs parlor comes on, (or until someone figures it out for me), I'll have to stick to my hardware solution.

Which is:



The four least significant bits are latched by a 62 instruction. That output feeds the 7485 and is compared with the keyboard input to the 7485. A match outputs a HI to the 7400. Another HI from an inverted keyboard strobe takes EF3 LO. This is the same thing the 4515 keyboard does---takes EF3 LO when the data on the data buss (during a 62 output) matches the output selected by the keyboard.

O.K! It would have been better to do it with software. So somebody show me!

THE THREE KEYBOARD PROBLEM (CONT'D)

At any rate, I went from 3 keyboards to one.

After I had torn up my ELF II keyboard, I wondered if the comparator circuit couldn't be used with it. Pin 12 of the 74C922 puts out a keyboard strobe. Could be worked into this circuit---I don't know whether the polarity would be right. Maybe somebody will try it and let me know.

Incidentally, there isn't much to moving the VIP operating system. Write it in RAM somewhere and write protect it. If you have 3K, you could write it at page 08 and 09 instead of 80 and 81 and you'll have enough room to play all 20 games in the book. "8001" should be changed to the page number of the FIRST page of your relocated op system. "8056" chould be changed to the SECOND page number of the relocated VIP op system. Then, in CHIP 8, at 000A, 010B, and 012A---replace the "81" with the page number of the SECOND page of the relocated operating system. That ought to VIP your ELF.....if the keyboard deal works.

PROGRAMMER'S ZODIAC

James W. Stephens
2324 Dennywood Dr.
Nashville, Tenn. 37214

DIRECTIONS: After entering all 256 instructions, push RST then Go. Then enter your birth month (01-12), into the keyboard, push IN, enter your birth day (01-31) into the keyboard, push IN, enter the hour of birth (01-24) for 24 hour day, push IN, Cosmac will tell you your true character (See Character Reference Sheet).

Register Assignement
- R5-month compare sub point
- RA-delay timer
- R1-X
- R6-month compare utility

(*) means that these locations or jump addresses are program placement sensitive and would be changed if program is moved in memory.

ADDRESS	INSTRUCTION	COMMENT
00	F8 00 B1 B5 B6 B4	Clear...
06	F8 E1* A4	mind of all
09	F8 FD* A1	previous desires.
0C	E1	
0D	F8 D6* A5	Collect thoughts
10	3F 10*	and all aims.
12	6C	
13	64 37 14*	Concentrate on...
16	3F 16*	stellar house.
18	6C	
19	64 37 1A*	Speak softly and
1C	3F 1C*	wait for signs...
1E	6C	
1F	64 37 20*	Consult the god
22	F8 FD* A1	of wisdom.... if
25	F8 01* A6	he's in and taking
28	D5 32 51* D5	visitors.
2C	32 5C* D5	
2F	32 67* D5	Send messenger

PROGRAMMER'S ZODIAC (CONT'D)

```
ADDRESS  INSTRUCTION          COMMENT
  32     32 72*D5             to mind of Zeus.
  35     32 7D*D5
  38     32 88*D5             Dismiss the guy
  3B     32 93*D5             for drinking on
  3E     32 9E*D5             job...
  41     32 A9*
  43     F8 10 A6 D5          Read subject's
  47     32 B4*D5             mind...
  4A     32 BF*D5
  4D     32 CA*               Read stars...
  4F     30 22*
  51     F8 21 F5             Read moon...
  54     3B CF*
  56     F8 F9*A1             Read "Gone with
  59     D4 30 22*            the Wind"
  5C     F8 20 F5
  5F     3B 56*               Make up something
  61     F8 FA*A1             for those who
  64     D4 30 22*            forgot birthday...
  67     F8 21 F5
  6A     3B 61*               Compare signs...
  6C     F8 FB*A1
  6F     D4 30 22*            Compare houses...
  72     F8 21 F5
  75     3B 6C*               Compare bank
  77     F8 FC*A1               books..
  7A     D4 30 22*
  7D     F8 22 F5
  80     3B 77                Ask for more
  82     F8 ED*A1             money.
  85     D4 30 22*
  88     F8 22 F5             Study the
  8B     3B 82*                   Cosmos..
  8D     F8 D5*A1
  90     D4 30 22*            Study the
  93     F8 24 F5                 Eros..
  96     3B 8D*
  98     F8 F5*A1             Study the
  9B     D4 30 22*                Koran..
  9E     F8 24 F5
  A1     3B 98*               Ask forgiveness
  A3     F8 11*A1
  A6     D4 30 22*            Take a nap..
  A9     F8 24 F5
  AC     3B A3*               and dream of
  AE     F8 21*A1               Atlantis
  B1     D4 30 22*
  B4     F8 24 F5             Blurt out a
  B7     3B AE*                 prediction
  B9     F8 4F*A1               or some
  BC     D4 30 22*              such drivel
  BF     F8 23 F5
  C2     3B B9*
```

PROGRAMMER'S ZODIAC (CONT'D)

```
ADDRESS  INSTRUCTION          COMMENT
   C4    F8 DA*A1             Waste a lot
   C7    D4 30 22*             of time..
   CA    F8 22 F5
   CD    3B C4*
   CF    F8 F8*A1
   D2    D4 30 22*            and repeat
   D5    D0
   D6    86 F5                for
   D8    3A DD*
   DA    60 30 D5*            those
   DD    16 30 D5*
   E0    D0                   that are
   E1    64
   E2    F8 6A BA
   E5    2A 9A                hard-of-
   E7    32 EB*
   E9    30 E5*               hearing
   EB    F8 01*A1
   EE    64
   EF    F8 2A BA             Wait for...
   F2    2A 9A
   F4    32 E0*               next...
   F6    30 F2*
   F8    40 50 70 80 90       victim
   FD    00 00 00
```

CHARACTER REFERENCE SHEET

If the Cosmac can get it all together long enough to compute, it will flash out the code for your character after you enter your month, day, and time (01-24 for 24 hour-day) of birth. If you can't remember the exact time, enter your favorite number which usually works for most Horoscopes...May the stars be with you.

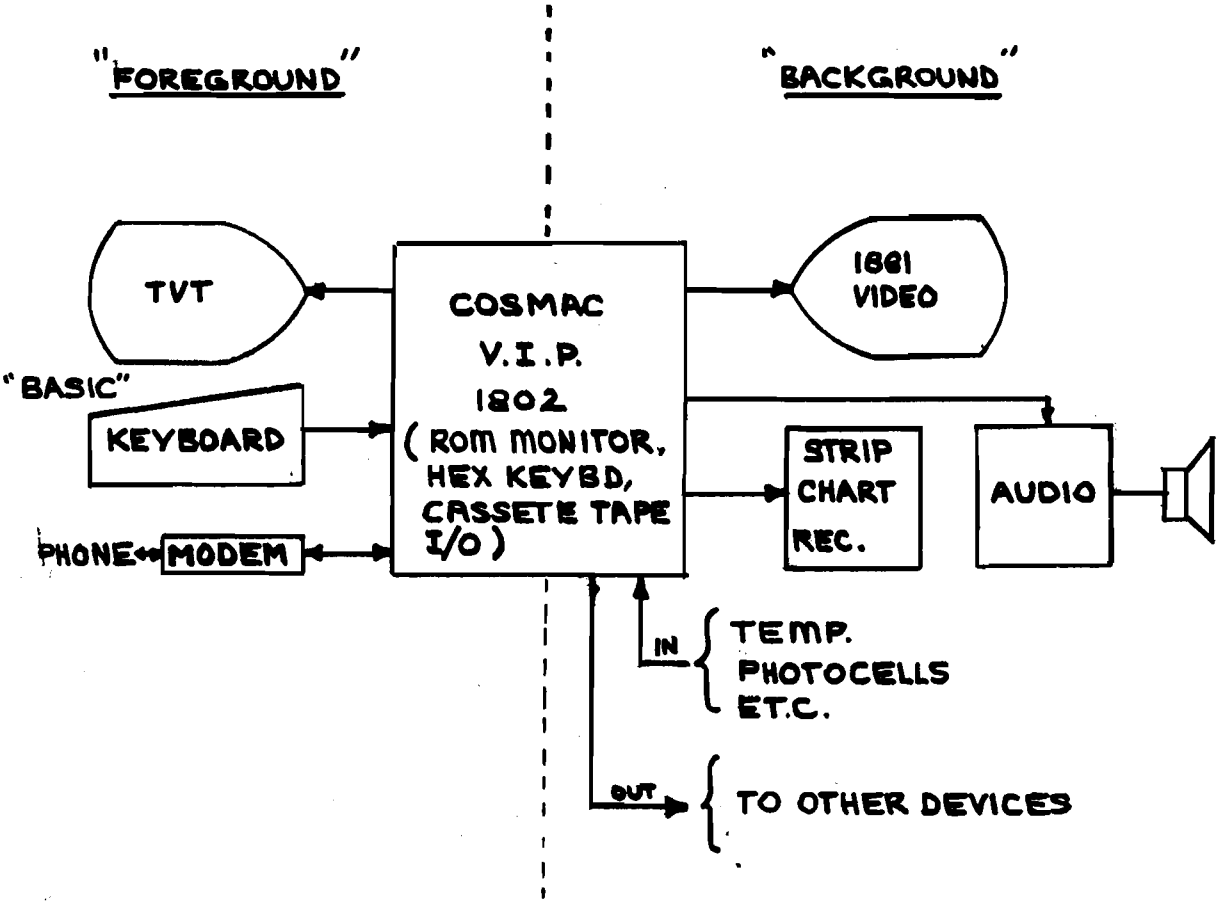
CODE	CHARACTER TRAITS ETC.
80	friendly; Idealistic; self-centered; fearless; honest; tempermental; this person sends 256 word hex programs to the Academy' of Computer Sciences, and a copy to Kilobaud.
90	solid and tranquil; seldom worries; deliberate and opinionated; usually thinks his computer system is best on the market and tells everyone in the club so.
A1	flighty; impatient; alert; neat; not puntual; talkative; this person writes short, neat programs and never debugs them.
D0	dreamy and immaginative; soft-hearted; secretive; generous; saves money to buy a HAL 2000 system without telling his wife.

CHARATER REFERENCE SHEET (CONT'D)

CODE	CHARACTER TRAITS ETC.
EO	egotistical; extroverted, astute; stoic; loyal; writes nasty letters to DDJ for not printing his articles.
10	worry-prone; methodical; meticulous; restless; careful; writes beautiful, expert, and original programs..then tears them up.
20	peaceful; good-natured; intelligent; naive'; indulgent; buys an Imsai with 65K, and disk and write game programs.
30	self-controlled; tactless; unforgiving; possessive; religious; wants to computerize the church and give each member a printout of their weekly rating.
60	free of malice; blunt; not graceful; restless; cheerful; wants to write programs but keeps getting tangled up in the cables and keyboard.
40	quiet; determined; ambitious; serious; adaptable; gloomy; studies computer programming and schematics to one day build the ultimate junk-box system, for less than $4.
50	realist; vague; friendly; futuristic; conceited; diplomatic; not precise; unpredictable; thinks computers will one day rule the world..or maybe the universe..or maybe...
70	freedom-loving; careless; deceptive; romantic; charming; non-competitive; helpful; dates four or five programmers who let him use the company's system at night.

AN RCA CPD1802 SYSTEN

Bill Freymuth
9119 E 66 St
Tulsa, Oklahoma 74133
(918) 252-5345



My system consists of a fully populated RCA COSMAC VIP (4K RAM plus I/O Port), ASCII Keyboard, bay area TVT (Electronic Systems, San Jose, CA., part no. 1C6) and several homebrew boards for I/O expansion, A to D and D to A. The first problem I encountered was exchanging tapes with Netronics ELF II users in my area. Tapes created by the ELF could be read by my VIP and vise versa, however, the data was all turned around. Depending on which system you have, the other writes the data bytes out backwards (ie, LSB first instead of MSB) and inverts each data bit. The program in figure 1 corrects this problem. This program may be executed by either the supplier of the data before writing it to tape or by the receiver after reading the data in (ie, the program will work on both the ELF and the VIP). Since the program runs in the first page of memory, any program that is to be converted which also normally runs in the

first page of memory will have to be loaded into a higher area for the conversion and then copied back to tape and read into the first page for execution. As indicated with the listing, the start and end addresses for the data must be provided. When the conversion is complete, the program sets Q and stops.

My next problem was getting the system to do something besides making pictures on a TV. I might, also, add that all of the following was done as inexpensively as possible using parts I already had (there are surely better ways, but not in my case). Figure 2 is a digital to analog converter (R/2R resistor network) coming off a latched output port. As indicated, I actually have two of these D to A's built, one to drive a voltage to frequency converter which then drives an audio amplifier for music and the other D to A to drive a strip chart recorder (0-1MA). With this nominal amount of hardware, a great deal of software reduction was possible. For example, in a music program all that is required is to sequentially process the data, first getting the note (Byte) and placing it on the latched output port and, then, looping for the length of time indicated by the duration (Byte) before getting the next note. As this is a simple program and the hardware is applicable to any system, I have only included a flow chart for the program in Figure 3. I might add that in order to set this up, the first thing that is required is a test program to place bytes on the output port and then determine the frequency generated by each byte. As indicated in Figure 2, the other D/A is used to drive a strip chart recorder, however, this could just as easily be used as input to either an analog or digital meter or for that matter any piece of hardware requiring a varying voltage and not too much current. The same program as the music program can be used for driving this.

Figure 4 is a schematic for a very inexpensive dual 8 bit analog to digital converter plus a digital on/off switch. I I intentionally left off some of the values as they will be a function of the particular application. In my case the clock input to the converter is set at approximately 500 HZ. This provides one update per second. Interfacing National's LX5600 temperature transducer chip (linear voltage to temperature output) to the two A to D's for indoor and outdoor temperature and a photo cell to the digital switch going to the flag line works nicely.

Although all of the hardware and software is not completed as yet, my system configuration is as follows. First, in the 'foreground' I will be running basic with serial input through a uart from the keyboard and parallel output to the TVT with computer cursor control. With this configuration relatively high speed graphics is still possible on the TVT. In the 'background', the 1861 video chip will still be functioning under interrupt for a second video output. The interrupt routine will not only handle the video refresh logic but will also maintain a real time clock, pick up indoor/outdoor temperature, play music and handle other control functions such as checking the flag line

with the photo cell on it. All of this will be displayed by the interrupt routine using the 1861. I might point out that all of this cannot be accomplished during one interrupt, therefore the routine is written such that each interrupt accomplishes a portion of the logic yet completes all of the required logic in less than one second (there are approximately 60 interrupts per second of which less than ten are required). Figure 5 is a simplified flow chart of this interrupt routine.

In addition to the above, I plan on adding a second uart connected to a modem (Electronic Systems Modem, part no. 109) for a tele-communications interface. This will be connected to an expansion module with six input and six output ports. Finally, I hope to have and S-100 interface to a 16K RAM board using 2114's and an EPROM board with 2708's or 2716's, along with an EPROM burner.

TAPE CONVERSION PROGRAM
ELF TO VIP - VIP TO ELF

0000	7A	F8	XX	BE	F8	XX	AE	F8
0008	XX	BF	F8	XX	AF	F8	00	B3
0010	F8	16	A3	30	21	D0	BA	8A
0018	FE	AA	9A	F6	33	15	1A	30
0020	15	F8	00	BB	F8	4B	AB	EB
0028	8F	73	9F	73	60	F8	00	AA
0030	0E	D3	D3	D3	D3	D3	D3	D3
0038	D3	8A	5E	9E	F5	3A	45	8E
0040	60	F5	32	48	2B	1E	30	2D
0048	7B	00	XX	XX	--	--	--	--

Starting Address for Data - 0002 (High Byte)
0005 (Low Byte)

Ending Address for Data - 0008 (High Byte)
000B (Low Byte)

The program uses locations 004A and 0048 for temporary storage.

FIGURE 1

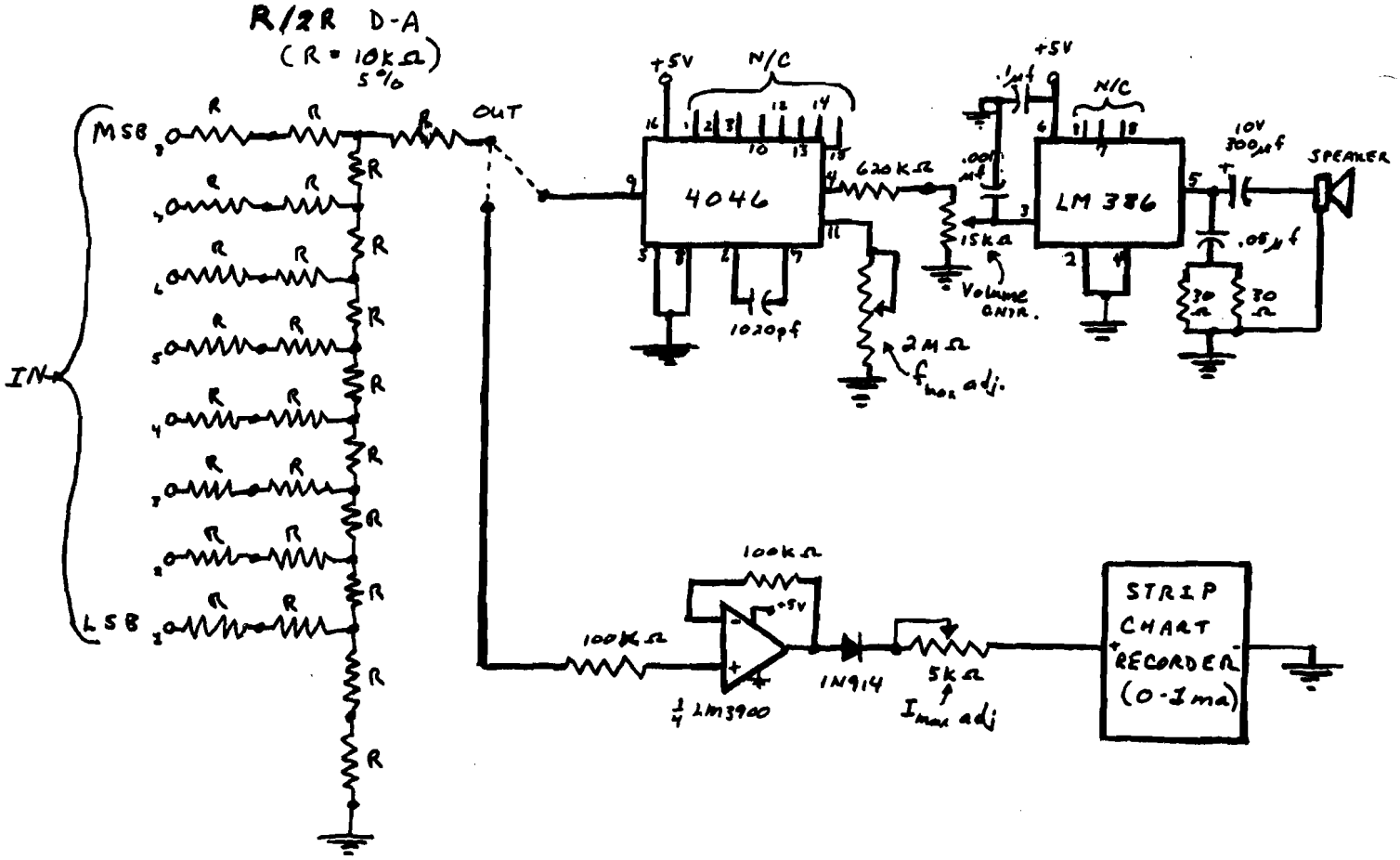


FIGURE 2

D to A CONVERTER

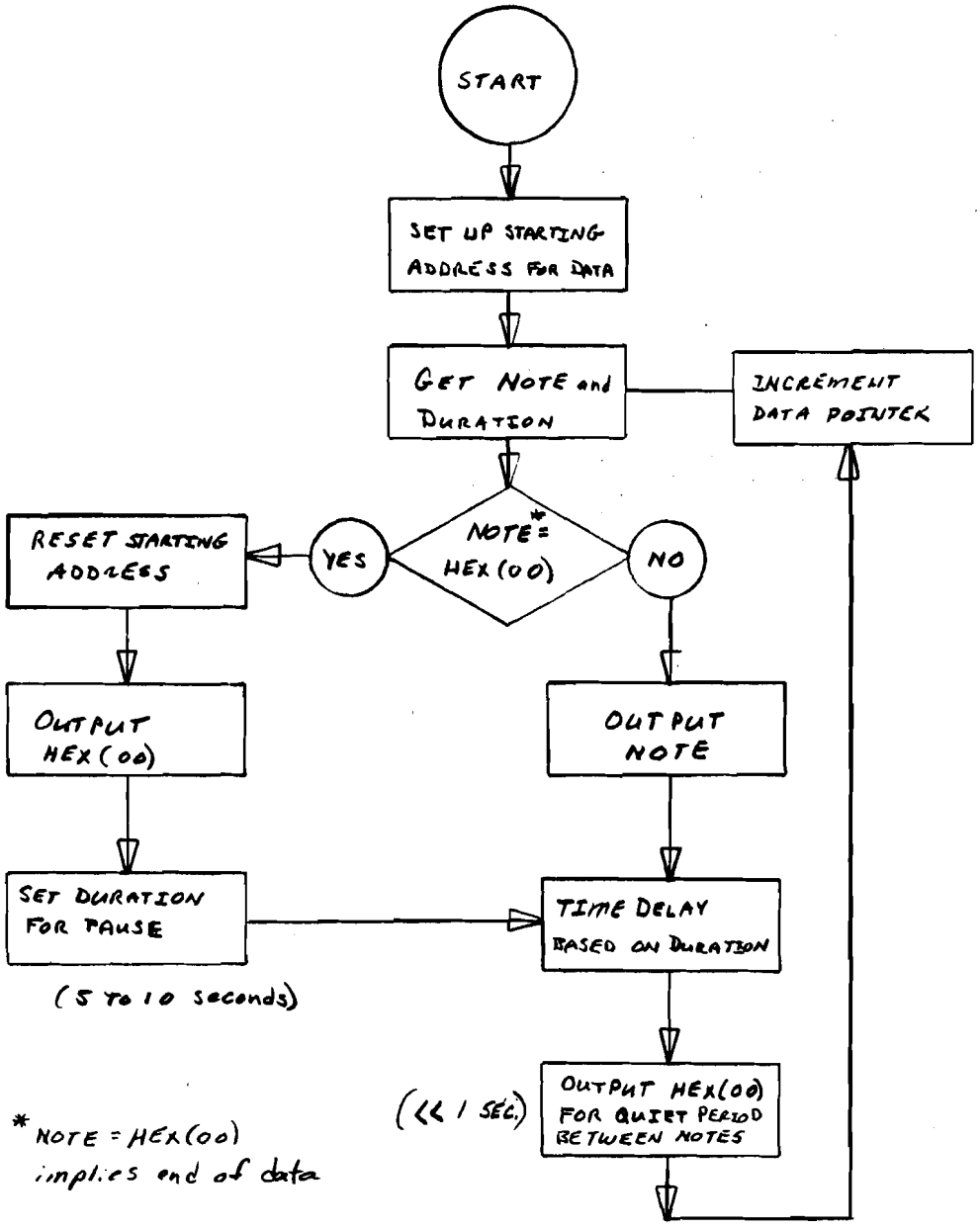


FIGURE 3

MUSIC PROGRAM FLOWCHART

FIGURE 4

DUAL A TO D WITH DIGITAL SWITCH

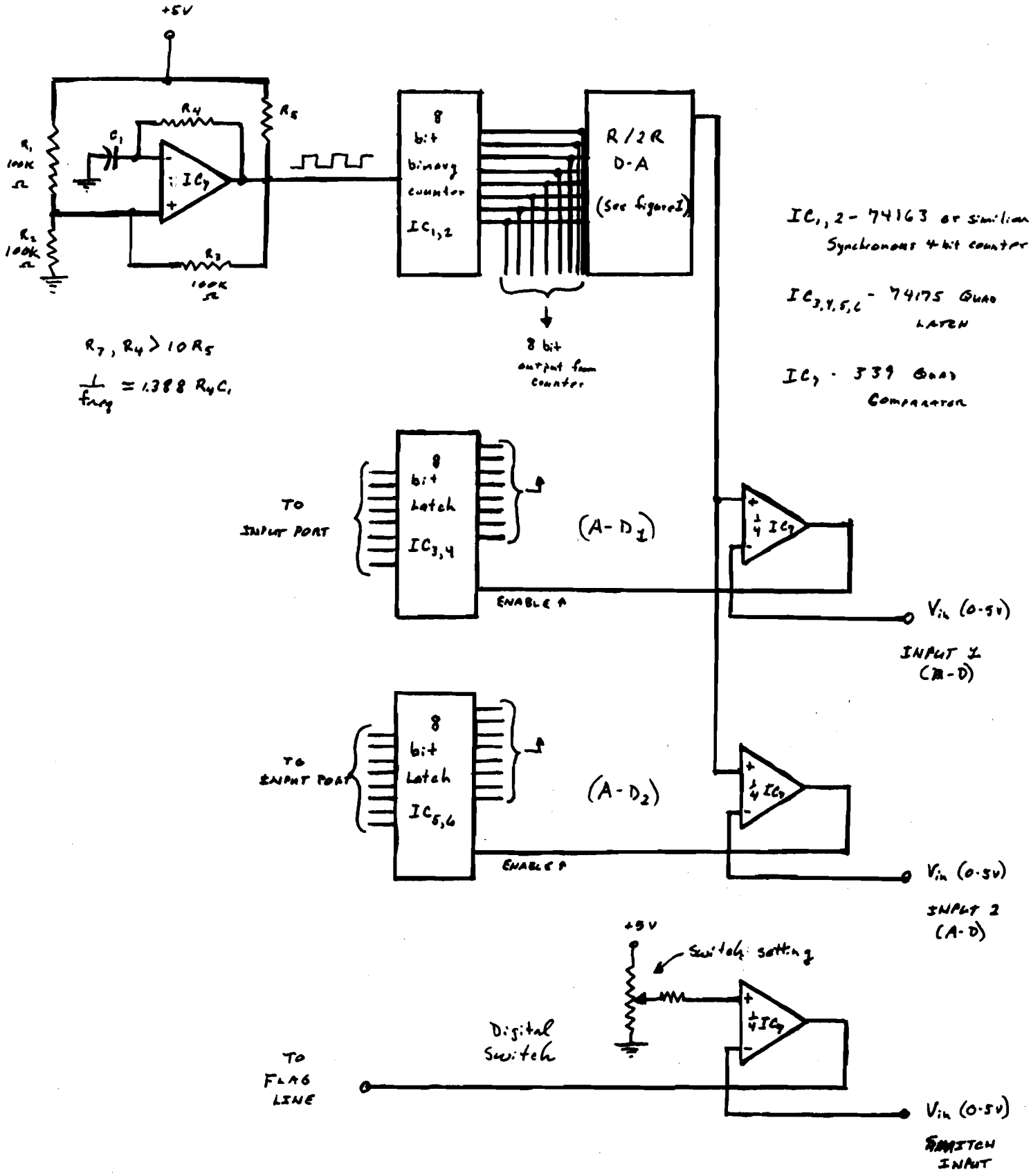
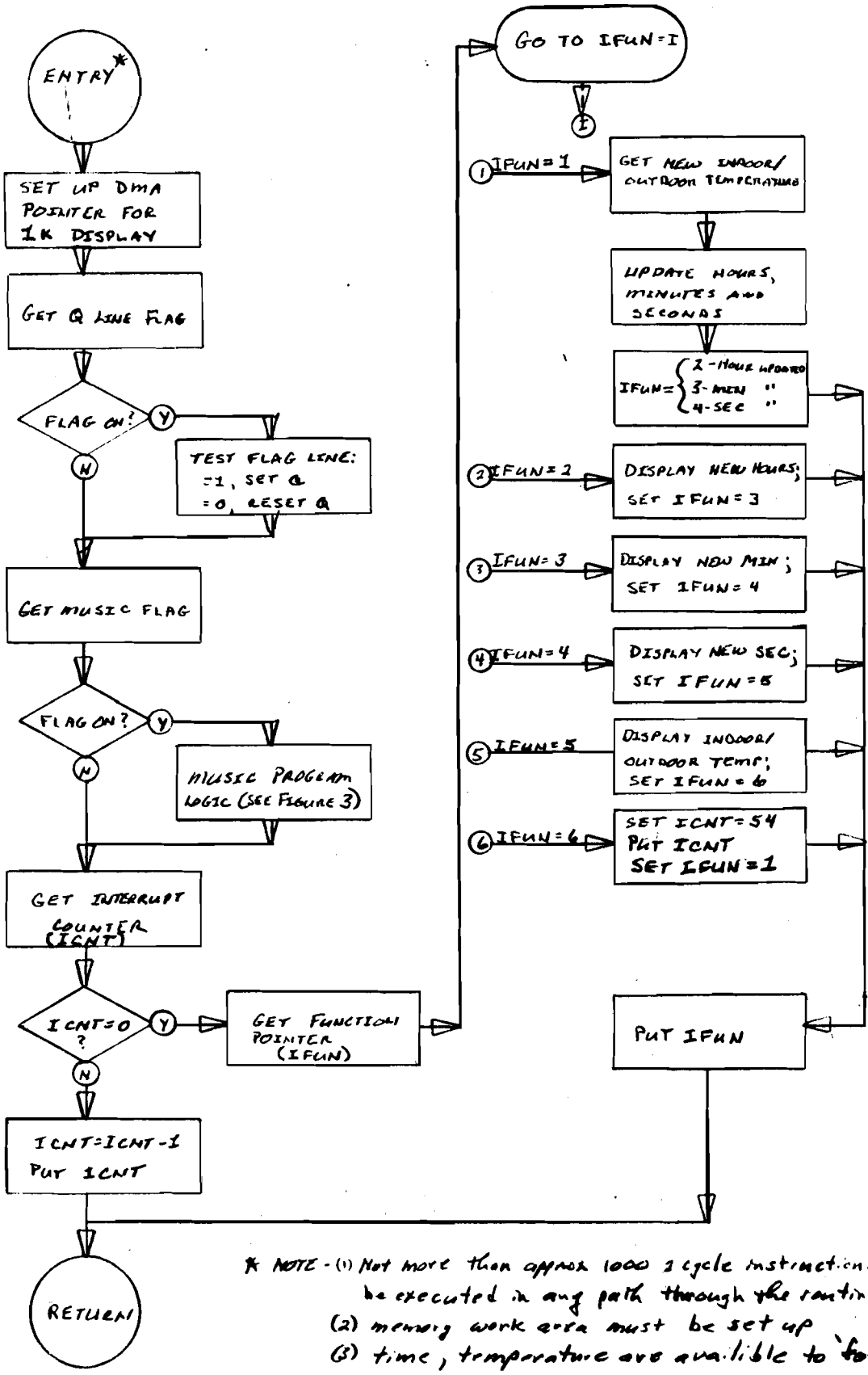


FIGURE 5

DISPLAY INTERRUPT ROUTINE



* NOTE - (1) Not more than approx 1000 1 cycle instructions should be executed in any path through the routine
(2) memory work area must be set up
(3) time, temperature are available to 'foreground'.

```
LOCN OBJ CODE  STMT    SOURCE STATEMENT                    1802 VER 1.6

                1 ..........................................................
                2 ..  THIS IS THE SOFTWARE FOR THE 1K VIDEO RAM ARTICLE
                3 ..  IPSO FACTO ISSUE #6, PAGE 29.
                4 ..........................................................
3800            5          ORG   #3800          .. START OF PROGRAM
3800      =3B60 6 BOTTOM:  EQU   #3B60          .. BOTTOM LINE OF DISPLAY
3800      =3A00 7 TOPLIN:  EQU   #3A00          .. TOP LINE OF DISPLAY
                8 ..
                9 .. THIS IS THE CHARACTER OUT ROUTINE. THE CHARACTER IS PASSED IN
               10 .. THROUGH D, THE ACCUMULATOR. CHARACTERS WITH THE SIGN BIT SET
               11 .. ARE IGNORED, CARRIAGE RETURN CAUSES THE POINTER TO BE RESET TO
               12 .. THE BEGINNING OF THE BOTTOM LINE OF THE DISPLAY (#3B60), AND
               13 .. LINE FEED CAUSES THE SCREEN TO BE SCROLLED UP AND THE POINTER
               14 .. TO BE RESET TO THE BEGINNING OF THE BOTTOM LINE.
               15 ..
3800 A0         16 CHROUT:  PLO   R0            .. HOLD CHAR FOR A SEC
3801 F838       17          LDI   A.1(KDATA)    .. SAVE ADDR FOR CHARACTER
3803 BE         18          PHI   R14
3804 F880       19          LDI   A.0(KDATA)
3806 AE         20          PLO   R14
3807 EE         21          SEX   R14
3808 80         22          GLO   R0            .. GET CHARACTER
3809 5E         23          STR   R14           .. NOW STORE
380A 327B       24          BZ    CONTRL        .. IGNORE CODE ZERO
380C FA80       25          ANI   #80           .. CHECK IF SIGN BIT
380E 3A7B       26          BNZ   CONTRL        .. IF ON, IGNORE
3810 F838       27          LDI   A.1(POS)      .. GET CURRENT LINE
3812 BF         28          PHI   R15           .. POSITION ADDR
3813 F87E       29          LDI   A.0(POS)      .. GET LOWER PART
3815 AF         30          PLO   R15           .. SAVE LOWER PART
3816 4F         31          LDA   R15           .. GET CURRENT POSITION
3817 B1         32          PHI   R1            .. PUT IT INTO R1
3818 0F         33          LDN   R15
3819 A1         34          PLO   R1
381A F0         35          LDX                 .. RESTORE CHARACTER
381B FF0A       36          SMI   #0A           .. LINE FEED?
381D 3244       37          BZ    SCROLL        .. YES, SCROOL SCREEN
381F F0         38          LDX                 .. RESTORE CHARACTER
3820 FF0D       39          SMI   #0D           .. C/R?
3822 3229       40          BZ    CR            .. BRANCH IF C/R
3824 F0         41 NOCR:    LDX                 .. RESTORE CHARACTER
3825 FF12       42          SMI   #12           .. CONTROL R?
3827 3A31       43          BNZ   NORET
3829 F83B       44 CR:      LDI   A.1(BOTTOM)   .. YES, RESET LINE POINTER
382B B1         45          PHI   R1
382C F860       46          LDI   A.0(BOTTOM)
382E A1         47          PLO   R1
382F 3076       48          BR    RETURN        .. DONE WITH THIS
3831 F0         49 NORET:   LDX                 .. RESTORE CHARACTER
3832 FF14       50          SMI   #14           .. CONTROL T?
3834 3A3C       51          BNZ   NOCT          .. BRANCH IF NOT
3836 60         52          IRX
3837 F801       53          LDI   #01
3839 73         54          STXD
383A 3076       55          BR    RETURN
383C F0         56 NOCT:    LDX                 .. RESTORE CHARACTER
383D 51         57          STR   R1            .. DISPLAY CHARACTER
383E 11         58          INC   R1            .. BUMP POINTER
```

```
LOCN OBJ CODE  STMT  SOURCE STATEMENT                    1802 VER 1.6

383F 81          59          GLO   R1            .. SEE IF END OF LINE HIT
3840 FF80        60          SMI   #80
3842 3A76        61          BNZ   RETURN        .. NO, JUST RETURN
3844 60          62 SCROLL:  IRX
3845 F0          63          LDX
3846 FF01        64          SMI   #01
3848 3A4D        65          BNZ   SC1A
384A 73          66          STXD
384B 3076        67          BR    RETURN
384D F83A        68 SC1A:    LDI   A.1(TOPLIN)   .. UP SCREEN
384F B0          69          PHI   R0            .. R0 POINTS TO TOP LINE
3850 B1          70          PHI   R1            .. R1 TO LINE BELOW THAT
3851 F800        71          LDI   A.0(TOPLIN)   .. FOR R0
3853 A0          72          PLO   R0
3854 F820        73          LDI   #20           .. FOR R1
3856 A1          74          PLO   R1
3857 41          75 SC1:     LDA   R1            .. MOVE ALL LINES UP 1
3858 50          76          STR   R0
3859 10          77          INC   R0
385A 90          78          GHI   R0
385B FF3B        79          SMI   A.1(BOTTOM)
385D 3A57        80          BNZ   SC1           .. NOT DONE YET?
385F 80          81          GLO   R0
3860 FF60        82          SMI   A.0(BOTTOM)
3862 3A57        83          BNZ   SC1           .. NOT DONE YET?
3864 F83B        84          LDI   A.1(BOTTOM)   .. DONE, RESET LINE POINTER
3866 B1          85          PHI   R1
3867 F860        86          LDI   A.0(BOTTOM)
3869 A1          87          PLO   R1
386A F820        88 BLANK:   LDI   #20           .. BLANK THE BOTTOM LINE
386C 51          89          STR   R1
386D 11          90          INC   R1
386E 81          91          GLO   R1
386F FF80        92          SMI   #80
3871 3A6A        93          BNZ   BLANK
3873 F860        94          LDI   A.0(BOTTOM)
3875 A1          95          PLO   R1
                 96 .. THIS IS WHERE TO COME WHEN DONE, STORE THE CURRENT LINE
                 97 .. POINTER AT POS
3876 81          98 RETURN:  GLO   R1            .. LOW PART OF POSITION
3877 5F          99          STR   R15
3878 2F         100          DEC   R15           .. HIGH PART
3879 91         101          GHI   R1
387A 5F         102          STR   R15
387B F0         103 CONTRL:  LDX                 .. RESTORE CHARACTER
387C E2         104          SEX   R2            .. RESTORE R2 AS X
387D D5         105          SEP   R5            .. RETURN TO BASIC
387E 3B         106 POS:     ,     #3B           .. START OF BOTTOM LINE
387F 60         107          ,     #60
3880 00         108 KDATA:   ,     #00           .. PLACE FOR KEYBOARD DATA
3881 00         109          ,     #00
3882            110          END

   0 DIAGNOSTICS GENERATED
  15 SYMBOLS

SYMBOL TABLE:
BOTTOM    3B60    TOPLIN    3A00    CHROUT    3800    NOCR      3824
CR        3829    NORET     3831    NOCT      383C    SCROLL    3844
SC1A      384D    SC1       3857    BLANK     386A    RETURN    3876
CONTRL    387B    POS       387E    KDATA     3880
```

VARIATIONS ON A THEME

Dave Hersker
Box 802
Conyngham, PA. 18219 USA

This is a variation on "The 1802 Music Machine" by Cec Williams which appeared in Issue 5 of IPSO FACTO. With this program you can play two diferent verses and repeat each of them any number of times before going on to the next verse. It also automatically returns to the first verse and starts over again ad infinitum (ad nauseum if you don't turn the program off). I've listed the entire program but have only commented the changes I've made. I've also included the "notes" for a few songs - more as I transcribe them.

Memory Loc	Object Code	Comments
00	E8	
01	F8 __ A4	1st Verse Counter
04	F8 __ A5	2nd Verse Counter
07	84	
08	32 10	
0A	24	
0B	F8 61 A8	1st Verse Starting Address
0E	30 17	
10	F8 __ A8	2nd Verse Starting Address
13	85	
14	32 01	
16	25	
17	F8 60 A9	New Speed Modifier loc'n
1A	72	
1B	AA	
1C	32 07	
1E	FA 70	
20	BB	
21	C4 C4 C4	
24	C4 C4 C4	
27	C4 C4 C4	
2A	8A	
2B	FA 0F	
2D	AC	
2E	F9 50	
30	AD	
31	0D	
32	AE	
33	8A	
34	FA 80	
36	32 3D	
38	8E	
39	F6	

Memory Loc'n.	Object Code
3A	FF 01
3C	AE
3D	9B
3E	BF
3F	9F
40	32 17
42	8E
43	FF 01
45	2F
46	3A 43
48	8C
49	32 3F
4B	31 50
4D	7B
4E	30 3F
50	7A
51	30 3F
53	FF
54	57
55	49
56	41
57	36
58	30
59	33
5A	5C
5B	52
5C	4D
5D	45
5E	3D
5F	39
60	C4
61	1st Music Byte

Load the hex number of repeats for each verse in the appropriate verse counter (loading 00 will skip that verse). Put the hex address in the 2nd verse starting address. Each verse must end with a 00 "note."

NEVER ON SUNDAY
(One Verse)

```
60      25 26 2C 26 2F 29 2B C4 A4 28 C4 A4 28 C4 A4
70   28 A4 AC 28 A4 49 49 49 49 20 25 26 2C 26 2F 29
80   28 C4 A4 28 C4 A4 28 C4 A4 28 29 28 2F 29 46 46
90   46 46 20 25 26 2D 26 2F 29 28 C4 A4 28 C4 A4 28
A0   C4 A4 28 A4 AC 28 A4 49 49 49 49 20 25 26 2D 26
B0   2F 29 28 C4 A4 28 C4 A4 28 C4 A4 28 29 28 2F 29
C0   46 46 46 46 40 C4 CC CD C6 C6 C6 C5 C6 AD A6 CF
D0   C6 CD CD CD CD CD AC AD C6 CD CC CC CC CC CC A4
E0   AC CD C4 49 49 49 49 40 C4 CC CD C6 C6 C6 C6 C6
F0   AD A6 CF C6 CD CD CD CD CD 40 00
```

Or if using original program load at Hex 50 and add the following instead of 40 00 listed at hex F9 and FA

```
   AC AD C6 CD CC CC CC CC CC A4 AC CD C4 46 46 46
   46 20 00
```

BABY ELEPHANT WALK
(Two verses - Memory Address of 2nd verse is hex AD)

```
60      3C 1E 29 AC AE AD AC AA 27 29 40 40 40 3C 1E
70   29 AC AE AD AC AA 49 49 49 AA 26 66 26 AA 26 2C
80   40 40 40 3C 1E 29 AC AE AD AC AA 27 29 40 40 20
90   29 CD CD AB 29 20 29 4C 4C 94 9C 94 19 2F 26 48
A0   48 1F 18 1F 1E 2C 2A 24 2C 40 40 40 00 3C 1E 29
B0   AC AE AD AC AA 27 29 40 40 40 3C 1E 29 AC AE AD
C0   AC AA 49 49 49 AA 26 46 46 46 26 AA 26 2C 40 40
D0   40 3C 1E 29 AC AE AD AC AA 27 29 40 40 20 29 CD
E0   CD AB 29 20 29 CC CC 95 9C 95 19 2F 26 48 48 1F
F0   18 1F 15 2C 2A 24 2C 40 40 40 00
```

To conserve memory space this program does not zero any of the high registers. If you load it and you get random notes run the following program (90 B1 B2 .... BF) and then key in the first sixteen bytes and try again. It works for me. I'd be glad to transcribe any requests if they're accompanied by the sheet music - my library is somewhat limited.

TINY BASIC SQUARE ROOT ROUTINE VERSION 2

Jim Howell
5472 Playa del Rey
San Jose, CA 95123

I would like to thank Tom Crawford for an excellent article on a Tiny BASIC square root routine in the August 1978 issue of IPSO FACTO. I have modified his "first try" routine so that it works and is simpler than Tom's "full blown" routine. The major difference between our routines is that my routine stops looping when the new trial value is one more than the old trial value (line 1060 below), as well as when they are equal (line 1050). If the branch in line 1060 is taken, either B or C could be the correct root: this is the reason for the test in line 1090.

```
1000  IF A>0 GOTO 1030
1010  B=0
1020  RETURN
1030  B=A/182+2
1040  C=(B+A/B)/2
1050  IF C=B RETURN
1060  IF C=B+1 GOTO 1090
1070  B=C
1080  GOTO 1040
1090  IF C*C<=A B=C
1100  RETURN
```

In the above, as in Tom's program, A is the number whose square root is to be found, B is the square root when the routine returns, and C is a temporary used for computing. Also, there is a typographical error in Tom's "first try" routine. The minus signs in lines 1050 and 1060 should be plus signs.

EXTREMELY FAST MEMORY TEST PROGRAM

Ken Smith
12 Sylvia Cres.
Hamilton, Ont. L8K 2A9

A memory test program appeared in issue #4 of "IPSO FACTO" by A. Dunlap. This is the one recommended by Tektron Corp. for testing their 7K RAM board. It takes on the order of 30 min. to test 7K using a 2MHZ clock. The exhaustiveness of a memory test program is a very debateable subject, because t ere are many "strange" problems that can occur in a defective RAM board. One noteable example is a single memory location changing after it is read. Once the RAM board is debugged initially, the RAM chips can often go bad at any time (usually during the debugging stage of a program). The memory test program presented here takes about 10 sec. to test 7K! (at 2 MHZ) This program is ideal for incorporating into a monitor ROM to test the RAM each time the system is turned on.

This test program uses pseudo-random number techniques. A PN sequence (Pseudo-Noise) is statistically random but the sequence is predictable since it is produced by finite-state logic. A PN sequence is written into RAM and it is read out and compared to the generated PN sequence. (The PN generator is reset to the same "seed" or starting point for read and write). The memory is read twice through; to see if the data is static. The write-read process is

EXTREMELY FAST MEMORY TEST PROGRAM (CONT'D)

repeated again, but the PN sequence is complemented so that each bit of RAM is tested for 1 and 0. There is no guarantee this program will find all possible faults (does any memory test?) but the probability is very small (but finite) that a fault won't be caught.

The program is page-relocatable, but in most cases it will start at 0000. Locations 0003, 0004 contains the HI, LO start address of RAM test. Locations 0005, 0006 contains the end address. In a typical 8K system, START = 008A FINISH = 3FFF.

To test RAM load program, reset run. If the Q comes on with no data leds on, the RAM is OK. If the data leds come on and Q on, this is the HI address of a bad location. Press "1" for the LO address. Press "1" again to find the next bad location. (PROGRAM ON PAGES 32 and 33).

AN 1802 DMA CONTROLLER

Ken Bevis
220 Cherry Post Drive
Mississauga, Ontario
L5A 1H9

The 1802 has a good DMA feature that can be used to dump and load data via a hardware UART. A simple software routine can be used to set register 0 to the start memory location---this is part of my monitor routine. I input the page and absolute location of memory that I wish to inspect followed by the page and absolute location where I wish DMA. When I enter the inspection routine, register 0 is then indexed.

The advantage of DMA is it runs independant of your current program by stealing cycles when signalled by the UART. The disadvantage is it requires a little more hardware (see schematic). This DMA interface contains a counter that sets the size of memory block. When used as DMA Out, all DMA action stops when the count is reached and an alarm sounds until the DMA OUT switch is restored. On DMA IN, the alarm sounds when the data input equals the block size setting. However, it will continue to receive if data is presented until the switch is restored.
(SCHEMATIC ON PAGE 34).

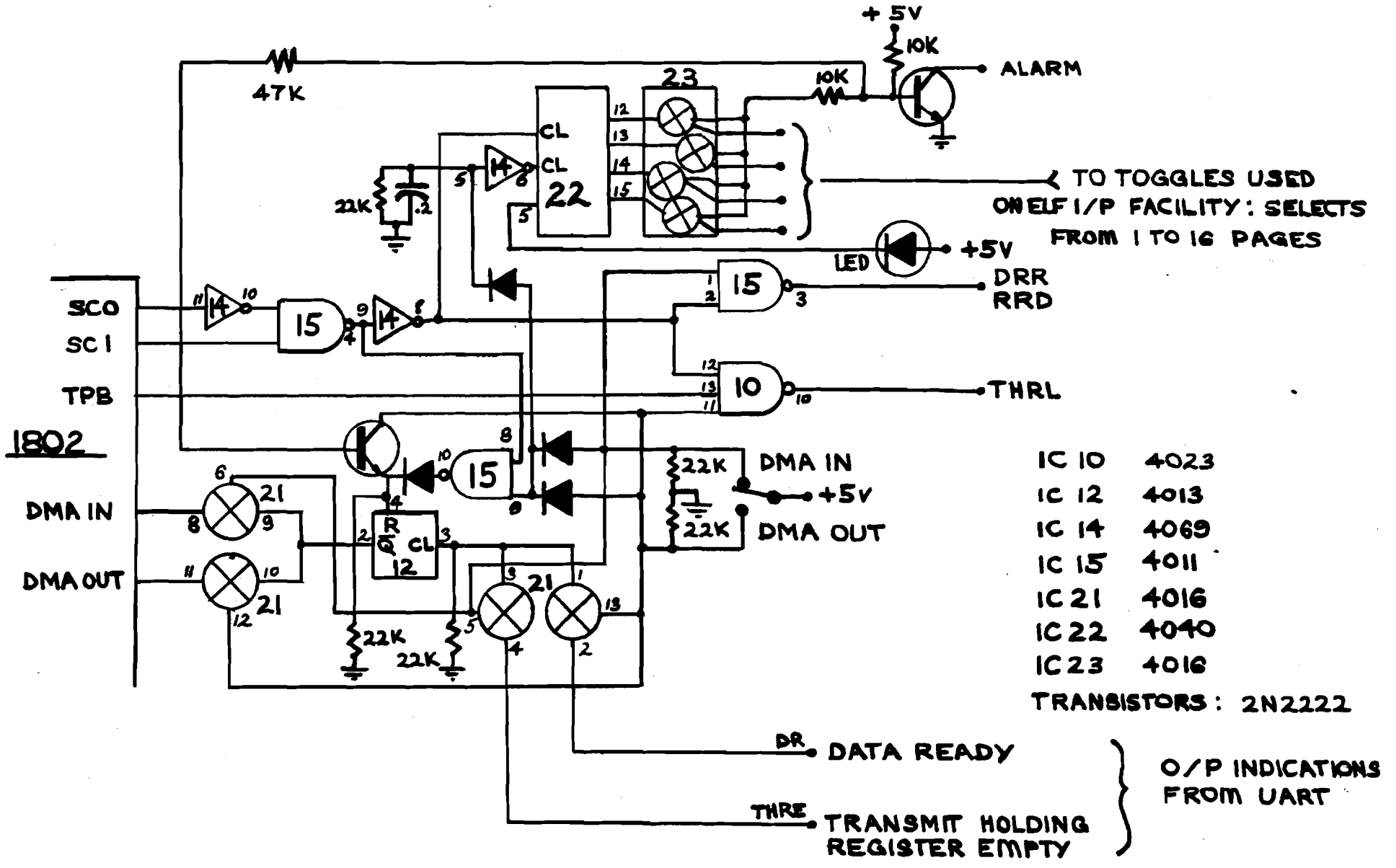
```
LOCN OBJ CODE  STMT  SOURCE STATEMENT                    1802 VER 1.6

                1 ..      SUPER FAST MEMORY TEST PROGRAM BY KEN SMITH
                2 ..      STANDARD RCA LEVEL 1 ASSEMBLY LANGUAGE SYNTAX
0000            3         ORG   #0000
0000 C4         4         NOP                  .. START HERE WITH PC=0
0001 3007       5         BR    AROUND         .. MISS DATA
0003 008A       6 START:  DC    A(#008A)       .. START ADDRESS OF RAM TEST
0005 1FFF       7 FINISH: DC    A(#1FFF)       .. END   ADDRESS OF RAM TEST
0007 90         8 AROUND: GHI   R0             .. INITIALIZE REGISTERS
0008 B2         9         PHI   R2             .. USING HI P.C. ALLOWS
0009 BA        10         PHI   RA             .. PAGE-RELOCATABILITY
000A F887      11         LDI   A.0(OUTLOC)    .. SET ADDRESS FOR SCRATCH
000C A2        12         PLO   R2
000D E2        13         SEX   R2
000E F800      14         LDI   #00            .. TURN OF DATA LEDS
0010 52        15         STR   R2
0011 64        16         OUT4
0012 22        17         DEC   R2
0013 A9        18         PLO   R9             .. R9.0 IS PN-DATA XOR MASK
0014 F803      19 RST:    LDI   A.0(START)     .. POINT TO MEMORY ADDRESS
0016 AA        20         PLO   RA             .. BOUNDARY DATA
0017 4A        21         LDA   RA             .. RE IS START ADDRESS
0018 BD        22         PHI   RD
0019 BE        23         PHI   RE
001A 4A        24         LDA   RA
001B AD        25         PLO   RD
001C AE        26         PLO   RE
001D 4A        27         LDA   RA             .. RF IS END ADDRESS
001E BF        28         PHI   RF
001F 4A        29         LDA   RA
0020 AF        30         PLO   RF
0021 F86A      31         LDI   A.0(PNGEN)     .. RA IS NOW P.C. FOR
0023 AA        32         PLO   RA             .. PN SUBROUTINE
0024 F8FF      33         LDI   #FF            .. FF IS "SEED", USE ANY BUT 0
0026 BC        34         PHI   RC             .. RC IS 15-BIT SHIFT REGISTER
0027 AC        35         PLO   RC
0028 DA        36 WRLOOP: SEP   RA             .. GENERATE PN BYTE
0029 9A        37         GHI   RA             .. GET PN BYTE
002A 5D        38         STR   RD             .. RD POINTS TO RAM LOCATION
002B 1D        39         INC   RD             .. POINT TO NEXT
002C 3B28      40         BNF   WRLOOP         .. TEST FOR END OF RAM
002E F802      41         LDI   #02            ..
0030 AB        42         PLO   RB             .. RB.0 IS READ LOOP COUNTER
0031 F8FF      43 RAGAIN: LDI   #FF            .. RESET PN GENERATOR
0033 BC        44         PHI   RC
0034 AC        45         PLO   RC
0035 9E        46         GHI   RE             .. RESET RAM START ADDRESS
0036 BD        47         PHI   RD
0037 8E        48         GLO   RE
0038 AD        49         PLO   RD
0039 DA        50 RDLOOP: SEP   RA             .. GENERATE PN BYTE
003A 4D        51         LDA   RD             .. COMPARE WITH VALUE
003B 52        52         STR   R2             .. WRITTEN IN RAM
003C 9A        53         GHI   RA
003D F3        54         XOR                  .. WITHOUT DISTURBING DF
003E 3A4E      55         BNZ   ERROR
0040 3B39      56 CHECK:  BNF   RDLOOP         .. TEST FOR END OF RAM
0042 2B        57         DEC   RB             .. READ THROUGH RAM TWICE
0043 8B        58         GLO   RB
```

```
LOCN OBJ CODE  STMT    SOURCE STATEMENT                    1802 VER 1.6

0044 3A31        59          BNZ   RAGAIN
0046 8A          60          GLO   RA          .. TEST IF DATA HAS BEEN
0047 3A85        61          BNZ   STOP        .. COMPLEMENTED
0049 F8FF        62          LDI   #FF         .. IF NOT, SET XOR-
004B AA          63          PLO   RA
004C 3014        64          BR    RST         .. MASK TO FF
004E 7B          65 ERROR:   SEQ               .. BAD NEWS
004F 2D          66          DEC   RD          .. POINT TO BAD LOCATION
0050 9D          67          GHI   RD
0051 52          68          STR   R2          .. DISPLAY HI-ADDRESS
0052 64          69          OUT4
0053 22          70          DEC   R2
0054 3754        71          B4    *           .. WAIT FOR "1"
0056 3F56        72          BN4   *
0058 8D          73          GLO   RD          .. DISPLAY LO-ADDRESS
0059 52          74          STR   R2
005A 64          75          OUT4
005B 22          76          DEC   R2
005C 375C        77          B4    *           .. WAIT FOR "1"
005E 3F5E        78          BN4   *
0060 1D          79          INC   RD          .. RESTORE RD
0061 7A          80          REQ
0062 F800        81          LDI   #00
0064 52          82          STR   R2
0065 64          83          OUT4
0066 22          84          DEC   R2
0067 3040        85          BR    CHECK       .. RESUME TESTING
0069 D0          86 EXIT:    SEP   R0          .. RETURN
006A 8C          87 PNGEN:   GLO   RC          .. GENERATE PN BYTE
006B 52          88          STR   R2
006C F6          89          SHR               .. XOR BIT 1 AND 2 OF RC
006D F3          90          XOR
006E F6          91          SHR
006F F6          92          SHR               .. SHIFT XOR RESULT INTO DF
0070 9C          93          GHI   RC          .. SHIFT RC RESULT
0071 76          94          SHRC              .. AND DF INTO BIT 15 OF RC
0072 BC          95          PHI   RC
0073 8C          96          GLO   RC
0074 76          97          SHRC
0075 AC          98          PLO   RC
0076 8F          99          GLO   RF         .. SUBTRACT RF FROM RD
0077 52         100          STR   R2
0078 8D         101          GLO   RD
0079 F7         102          SM
007A 9F         103          GHI   RF          .. SET DF=0 IF RD$RF
007B 52         104          STR   R2          .. PRESENT ADDRESS $ END ADDRESS
007C 9D         105          GHI   RD          .. SET DF=1 IF RD ŧ= RF
007D 77         106          SMB
007E 9C         107          GHI   RC          .. INVERT OR NON-INVERT PN RESULT
007F 52         108          STR   R2
0080 8A         109          GLO   RA
0081 F3         110          XOR               .. FINAL PN BYTE IS IN R9.1
0082 B9         111          PHI   R9
0083 3069       112          BR    EXIT
0085 7B         113 STOP:    SEQ               .. MEMORY TEST COMPLETE
0086 00         114          IDL
0087 00         115 OUTLOC:  DC    #00
0088            116          END

  0 DIAGNOSTICS GENERATED
 13 SYMBOLS
```


+5V
10K
ALARM
47K
23
10K
CL
CL
22
22K
.2
TO TOGGLES USED
ON ELF I/P FACILITY: SELECTS
FROM 1 TO 16 PAGES
LED
+5V
SCO
SC1
TPB
1802
15
DRR
RRD
10
THRL
DMA IN
DMA OUT
22K
DMA IN
+5V
DMA OUT
22K
R
Q
CL
12
21
22K
22K
DR
DATA READY
THRE
TRANSMIT HOLDING
REGISTER EMPTY
O/P INDICATIONS
FROM UART
IC 10 4023
IC 12 4013
IC 14 4069
IC 15 4011
IC 21 4016
IC 22 4040
IC 23 4016
TRANSISTORS: 2N2222

MEMORY PAGE DECODERS

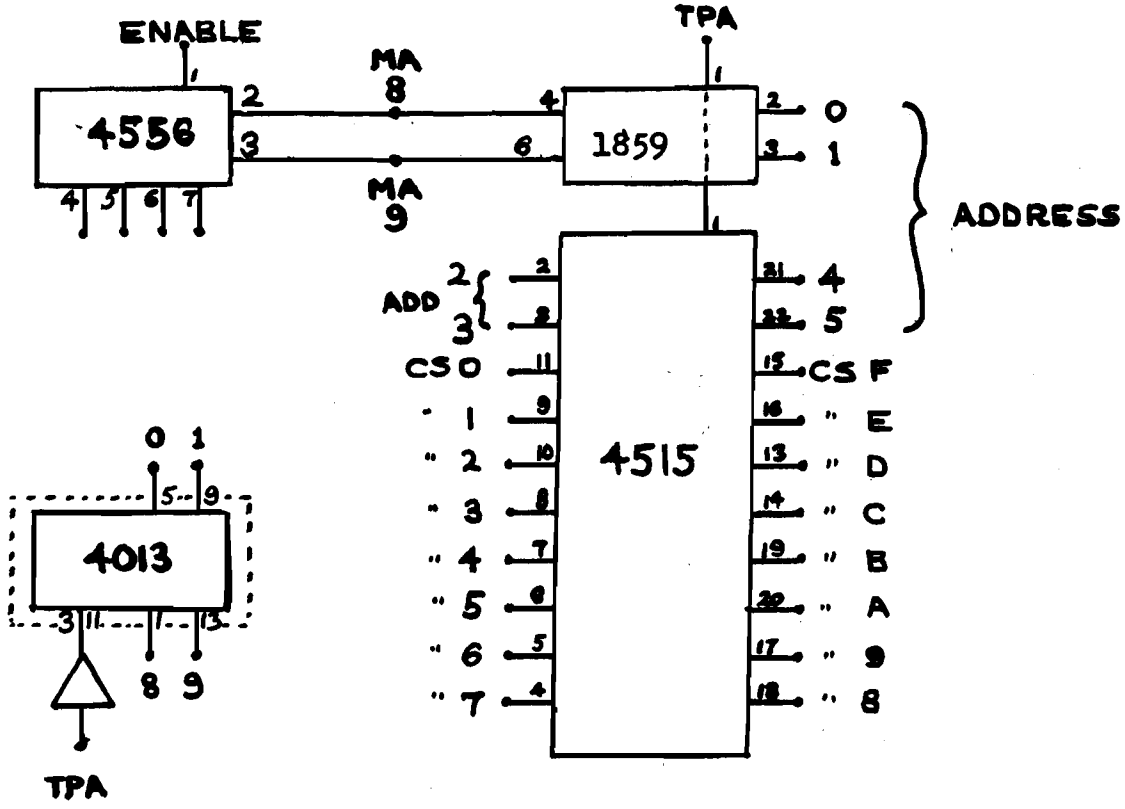
Ken Bevis
220 Cherry Post Drive
Mississauga, Ontario
L5A 1H9

When additional memory is added to the 1802, further page decoding is necessary. The following schematic describes a minimum system for decoding up to 16K of 2102 or 4116 chips. If you have smaller than 1K chips, this is handled by further breaking out a 1K block to its 4 pages.

The CMOS 4515 is a 4 line to 16 line decoder complete with latch. The latch is triggered on the trailing edge of TPA which allows full time for the high order address bits to stablize which means no problem exceeding 2 or 3 megahertz clock.

The CMOS 4555 is enabled by the 4515 in the situation where 1K must be broken out into 4 pages; two of these circuits are available on the one chip.

The third chip is the RCA CDP 1859 which is used to latch address leads 8 and 9. If this chip cannot be located, a D FLOP and single inverter can be substituted.



LOGIC PROBE

M.J. Pupeza
644 Bathurst St.
Toronto, Ontario

One of the easiest to use and most useful trouble shooting aids for problems in your micro system is a good logic probe. After playing around with quite a few circuits, I finalized on this relatively simple circuit modified from one lifted from Popular Electronics March 1974.

I made several mods, mainly to increase the input impedance to not load down CMOS circuits excessively.

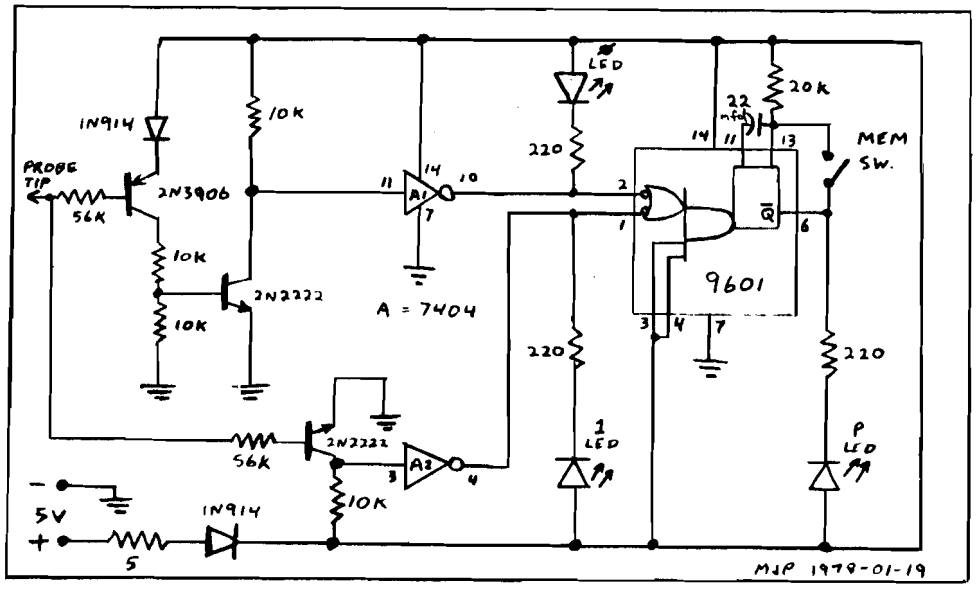
The circuit actually works best on TTL since the leds turn on at +2.6V and below +0.8V. A floating input does not turn on either led and any change of state causes the pulse led to come on for a moment. A pulse train causes the pulse led to remain lit and will record single pulses down to a 50 ns one.

The memory switch allows the pulse led to come on and stay on at any change in state of the input. This is handy when tracing a single narrow pulse that is not repetitive and can also be used to detect power interruptions.

For CMOS applications, this probe will detect the problem almost all the time if the power supply is at 5 Volts for the circuits and the probe. The only time you might fail to get a proper indication on the probe is when the signals do not reach the full high or low potentials due to loading. This hasn't happened to me yet.

Construction of my model was on some perf board and slipped into a short length of 1" copper tubing in which I had drilled 3 holes for the red, green, and amber leds. I installed a test probe into a plastic crutch tip and slipped it on one end and the switch and short length of coax power cable on a crutch tip slipped onto the other end. Any type of holder such as a penlite case, a cigar case, or similar could be used.

This probe works very well (a lot better than most I've tried) and has become indispensible for work on my system and other projects. While it can't replace a good scope, it certainly beats my Radio Shack $9 multimeter.



4.7V

LEVEL
HI > 2.6 V
LO < .8 V

PROBE LOAD
HI ≈ .07ma
LO ≈ .04ma

SOFTWARE FOR THE "IPSO FACTO" STANDARD FOR KANSAS CITY TAPES

Bernie Murphy
102 McCraney St.
Oakville, Ontario
Canada, L6H 1H6

The Association of Computer Experimenters has formally adopted the software standard as described in IPSO FACTO Issue #6, page 5.

THE SIMPLIFIED OR "S" FORMAT:

I) At least 15 seconds of mark (2400 HZ) at the beginning of a file. (we could make this lower)

II) All data bytes are 8 bits plus one start bit, even parity and one or more stop bits.

III) The Simplified format as discussed in the third issue of IPSO FACTO be used but slightly modified.

- file type	1 byte	ASCII character upper case S	
- start address high	1 byte	load address high	
- start address low	1 byte	load address low	
- byte count	2 bytes	16 bit length (hi then lo byte)	
- data byte 1	1 byte	8 bits	n x the value of the byte count
.	.	.	
.	.	.	
.	.	.	
- data byte n	1 byte	8 bits	
- checksum	1 byte	8 bits of checksum formed by adding all bytes excluding file type & checksum byte then, taking the ones complement of the above sum.	

WHY THIS ONE?

The above standard was adopted because the software to implement it is very minimal. We will call this standard the S or Simplified format. As can be seen by examining the listing, the whole thing easily fits in 256 bytes of memory.

WHY USE ONES COMPLEMENT IN THE CHECKSUM?

This method was "lifted" from Motorola's MIKBUG monitor for the 6800. Using this method, the cassette read routine can easily determine whether the computed checksum and the one read from the tape are identical (see lines 52 to 55 in the listing). Clever? Yes!

THE DELAY ROUTINE (lines 120 to 127)

If you have a "true" KANSAS CITY hardware interface, you will probably want to use the recovered clock instead of the CPU clock to clock in the data bits. See IPSO FACTO Issue #3 pages 23-30 for details. (CONTINUED ON PAGE 43)

```
LOCN OBJ CODE  STMT    SOURCE STATEMENT                    1802 VER 1.6

                    1 * * * * * * * * * * * * * * * * * * * * * * * * *
                    2 *         CASSETTE TAPE LOADER          OCT 10/78
                    3 *
                    4 *         REGISTER USAGE:     R0-R1=FREE
                    5 *                             R2=STACK
                    6 *                             R3=INITIAL P.C.
                    7 *                             R8=BYTE COUNT
                    8 *                             R9.0=CHECKSUM
                    9 *                             R10=PARITY COUNTER
                   10 *                             R11=LOAD ADDR
                   11 *                             R12=DELAY WORK
                   12 *                             R13=RX SUBR P.C.
                   13 *                             R14=SUBR WORK
                   14 *                             R15=DELAY P.C.
                   15 *
                   16 * * * * * * * * * * * * * * * * * * * * * * * * *
0000       =0009   17 CHECKREG EQU   9               CHECKSUM REG
0000       =00D0   18 STACK    EQU   $00D0           RAM WORK AREA
0000 93            19 LOADSTRT GHI   R3              CURRENT PAGE
0001 BD            20          PHI   R13             PUT IN R13 HIGH
0002 BF            21          PHI   R15             PUT IN R15 HIGH
0003 F800          22          LDI   A.1(STACK)      RAM ADDR
0005 B2            23          PHI   R2              A.1 OF STACK
0006 F8D0          24          LDI   A.0(STACK)      A.0 OF STACK
0008 A2            25          PLO   R2              INTO R2
0009 F833          26          LDI   RXENTRY         ADDR OF READ ROUT
000B AD            27          PLO   R13             INTO R13
000C F866          28          LDI   DELAY           SET UP R15 AS DELAY
000E AF            29          PLO   R15             SUBROUTINE P.C.
000F DD            30 HEADER   SEP   R13             LOOK FOR HEADER
0010 FF53          31          SMI   $53             UPPER CASE S ?
0012 3A0F          32          BNZ   HEADER          BRANCH IF NOT
0014 A9            33          PLO   CHECKREG        ZERO CHECKSUM
0015 7A            34          REQ                   RESET Q
0016 DD            35          SEP   R13             GET ADDR HIGH
0017 BB            36          PHI   R11             SAVE
0018 DD            37          SEP   R13             GET ADDR LOW
0019 AB            38          PLO   R11             SAVE
001A DD            39          SEP   R13             GET BYTE COUNT HIGH
001B B8            40          PHI   R8              AND SAVE
001C DD            41          SEP   R13             GET BYTE COUNT LOW
001D A8            42          PLO   R8              AND SAVE ALSO
001E DD            43 LOADLOOP SEP   R13             READ DATA BYTE
001F EB            44          SEX   R11             SET UP DATA POINTER
0020 5B            45          STR   R11             STORE IT
0021 64            46          OUT4                  DISPLAY & INC R11
0022 28            47          DEC   R8              DEC BYTE COUNT
0023 88            48          GLO   R8              LOAD NEW COUNT
0024 3A1E          49          BNZ   LOADLOOP        DONE YET ?
0026 98            50          GHI   R8              GET COUNT HIGH
0027 3A1E          51          BNZ   LOADLOOP        DONE YET ?
0029 DD            52          SEP   R13             GET CHECKSUM FROM TAPE
002A 19            53          INC   CHECKREG        ADD 1 TO ONES COMPLEMENT
002B 89            54          GLO   CHECKREG        GET CHECK SUM
002C 3A2C          55          BNZ   *               OH-NO...
002E 3000          56          BR    LOADSTRT        ANOTHER FILE
0030 C4            57          NOP                   FOR PATCH
0031 C4            58          NOP                   FOR PATCH
```

```
LOCN OBJ CODE  STMT    SOURCE STATEMENT                    1802 VER 1.6

                 59 * * * * * * * * * * * * * * * * * * * * * * * * *
                 60 *        CASSETTE INPUT ROUTINE
                 61 *        USES FLAG 2 AS SERIAL INPUT PORT
                 62 *
                 63 *        REGISTER USAGE :    R15=PC FOR DELAY SUBR
                 64 *                            R14=CHAR,BIT COUNTER
                 65 *                            R13=PC FOR ROUTINE
                 66 *                            R10=PARITY BIT COUNTER
                 67 *
                 68 *        INPUTS:             FLAG EF2
                 69 *
                 70 *        OUTPUTS:            RESULT IN D REG
                 71 *
                 72 * * * * * * * * * * * * * * * * * * * * * * * * *
0032 D3          73 RXRETURN SEP   R3            RETURN TO CALLER
0033 F808        74 RXENTRY  LDI   8             EXPECT 8 BITS
0035 AE          75          PLO   R14           SAVE AWAY
0036 3D36        76 RXSTART  BN2   *             WAIT FOR START BIT
0038 F81F        77          LDI   31            1/2 BIT TIME
                 78 * NOTE:  ABOVE VALUE WORKS ONLY FOR 1 MHZ CLOCK
003A 1F          79          INC   R15           R15=R15+2 TO MISS
003B 1F          80          INC   R15           LDI 62 AT °DELAY°
003C DF          81          SEP   R15           DELAY 1/2 BIT TIME
003D 3D36        82          BN2   RXSTART       GOOD START BIT?
003F F800        83          LDI   0             ZERO BIT COUNTER
0041 AA          84          PLO   R10           NOW...
0042 DF          85 RXLOOP   SEP   R15           DELAY ONE BIT TIME
0043 FC00        86          ADI   0             RESET DF
0045 354A        87          B2    RXBITOFF      BIT ON ?
0047 FF00        88          SMI   0             SET DF
0049 1A          89          INC   R10           COUNT BIT
004A 9E          90 RXBITOFF GHI   R14           GET CHAR
004B 76          91          SHRC                SHIFT DF INTO CHAR
004C BE          92          PHI   R14           SAVE RESULT TILL NOW
004D 2E          93          DEC   R14           NUMBER OF BITS TO GO
004E 8E          94          GLO   R14           INTO D
004F 3A42        95          BNZ   RXLOOP        DONE YET ?
0051 DF          96          SEP   R15           DELAY FOR PARITY BIT
0052 3555        97          B2    NOPARITY      PARITY BIT ON?
0054 1A          98          INC   R10           YES, INCREMENT
0055 DF          99 NOPARITY SEP   R15           DELAY FOR STOP BIT
0056 E2         100          SEX   R2            SET UP STACK
0057 89         101          GLO   CHECKREG      CHEKSUM TILL NOW
0058 52         102          STR   R2            STORE CHECKSUM
0059 9E         103          GHI   R14           GET CURRENT CHARACTER
005A F4         104          ADD                 DO CHECK SUM ADD
005B A9         105          PLO   CHECKREG      SAVE AWAY
005C 8A         106          GLO   R10           PARITY BIT COUNT
005D 76         107          SHRC                LSB SHOULD BE ZERO
005E 9E         108          GHI   R14           GET OUR CHARACTER
005F 7A         109          REQ                 ASSUME NO PARITY ERR
0060 3B32       110          BNF   RXRETURN      NONE
0062 7B         111          SEQ                 BAD NEWS...
0063 3032       112          BR    RXRETURN      RETURN TO CALLER
```

```
LOCN OBJ CODE  STMT    SOURCE STATEMENT                    1802 VER 1.6

               ..3 * * * * * * * * * * * * * * * * * * * * * * * * * *
               114 *        DELAY ROUTINE
               115 *
               116 *        REGISTER USAGE:     R13=CALLER P.C.
               117 *                            R12.0=WORK REG
               118 *
               119 * * * * * * * * * * * * * * * * * * * * * * * * * *
0065 DD        120 DELRTN   SEP   R13           RETURN TO CALLER
0066 F83E      121 DELAY    LDI   62            1 BIT TIME AT 1 MHZ
               122 * NOTE:  ABOVE VALUE WORKS ONLY FOR 1 MHZ CLOCK
0068 AC        123          PLO   R12           SAVE AWAY
0069 2C        124 DELDECR  DEC   R12           COUNT DOWN
006A 8C        125          GLO   R12           RESULT INTO D
006B 3A69      126          BNZ   DELDECR       DONE ?
006D 3065      127          BR    DELRTN        RETURN TO CALLER
               128 *
               129 *
               130 * * * * * * * * * * * * * * * * * * * * * * * * * *
               131 *        CASSETTE DUMP ROUTINE
               132 *
               133 *        REGISTER USAGE:     R2=STACK
               134 *                            R3=P.C.
               135 *                            R7.1=INPUT PAGE
               136 *                            R7.0=0
               137 *                            R8.1=OUTPUT PAGE
               138 *                            R9=LENGTH OF WRITE
               139 *                            R10.0=PARITY BIT COUNTER
               140 *                            R11.0=CHECKSUM REGISTER
               141 *                            R13=TX SUBR P.C.
               142 *                            R14=HEXKEYIN P.C.
               143 *                            R15=DELAY P.C.
               144 *
               145 * * * * * * * * * * * * * * * * * * * * * * * * * *
006F E2        146 DUMPSTRT SEX   R2            STACK IS R2
0070 93        147          GHI   R3            GET CURRENT PAGE
0071 BD        148          PHI   R13           PUT IN R13.1
0072 BE        149          PHI   R14           PUT IN R14.1
0073 BF        150          PHI   R15           PUT IN R15.1
0074 F8D0      151          LDI   A.0(STACK)    SET UP R2
0076 A2        152          PLO   R2            AS THE
0077 F800      153          LDI   A.1(STACK)    STACK
0079 B2        154          PHI   R2            POINTER
007A F8B0      155          LDI   TXENTRY       ADDR OF TXOUT
007C AD        156          PLO   R13           SAVE IN R9
007D F8D9      157          LDI   KEYENTRY      SET UP R14 AS KEYIN P.C.
007F AE        158          PLO   R14           SUBROUTINE P.C.
0080 F866      159          LDI   DELAY         SET UP R15 AS DELAY
0082 AF        160          PLO   R15           SUBROUTINE P.C.
               161 *
0083 7B        162          SEQ                 INDICATE INPUT PAGE
0084 DE        163          SEP   R14           CALL HEXKEY IN
0085 B7        164          PHI   R7            SAVE MEMORY PAGE
               165 *
0086 7A        166          REQ                 INDICATE OUTPUT PAGE
0087 DE        167          SEP   R14           CALL HEXKEY IN
0088 B8        168          PHI   R8            SAVE OUTPUT PAGE
               169 *
0089 7B        170          SEQ                 INDICATE LENGTH AND MARK
```

```
008A DE          171           SEP   R14          CALL HEXKEY IN
008B B9          172           PHI   R9           NUMBER OF PAGES
                 173 *
008C F800        174           LDI   0            D=0
008E A7          175           PLO   R7           R7.0=0
008F A9          176           PLO   R9           R9.0=0
0090 F853        177           LDI   $53          HEADER IS °S°
0092 DD          178           SEP   R13          WRITE OUT HEADER
0093 87          179           GLO   R7           D=0
0094 AB          180           PLO   R11          CLEAR CHECKSUM REG
0095 98          181           GHI   R8           GET OUTPUT PAGE
0096 DD          182           SEP   R13          WRITE ADDR HIGH
0097 87          183           GLO   R7           ADDR LOW=0
0098 DD          184           SEP   R13          WRITE ADDR LOW
0099 99          185           GHI   R9           LENGTH HI
009A DD          186           SEP   R13          WRITE LENGTH HIGH
009B 89          187           GLO   R9           LENGTH LOW
009C DD          188           SEP   R13          WRITE LENGTH LOW
                 189 *
009D E7          190 WRITDATA  SEX   R7           X=ADDR(INPUT PAGE)
009E 07          191           LDN   R7           GET DATA BYTE
009F 64          192           OUT4               DISPLAY & INC R7
 ⌄A0 DD          193           SEP   R13          WRITE DATA BYTE
00A1 29          194           DEC   R9           DECREMENT LENGTH
00A2 89          195           GLO   R9           GET LOWER COUNTER
00A3 3A9D        196           BNZ   WRITDATA     DONE A PAGE YET?
00A5 99          197           GHI   R9           GET HIGHER COUNTER
00A6 3A9D        198           BNZ   WRITDATA     DONE ALL PAGES YET?
00A8 8B          199           GLO   R11          GET CHECK SUM
00A9 FBFF        200           XRI   $FF          ONES COMPLEMENT
00AB DD          201           SEP   R13          WRITE IT
00AC 00          202           IDL                QUIT
00AD C4          203           NOP                FOR PATCH
00AE C4          204           NOP                FOR PATCH
```

```
LOCN OBJ CODE  STMT    SOURCE STATEMENT                    1802 VER 1.6

               205 * * * * * * * * * * * * * * * * * * * * * * * * * *
               206 *         CASSETTE OUTPUT ROUTINE
               207 *
               208 *         REGISTER USAGE :      R15=PC FOR DELAY SUBR
               209 *                               R14=CHAR,BIT COUNTER
               210 *                               R13=CHAR INPUT
               211 *                               R12=LOOP COUNTER
               212 *                               R11=CHECKSUM REGISTER
               213 *                               R10=PARITY BIT COUNTER
               214 *
               215 *         INPUTS:               D=CHARACTER TO BE
               216 *                               SENT
               217 *
               218 *         OUTPUTS:              Q LINE...SERIAL OUT
               219 *
               220 * * * * * * * * * * * * * * * * * * * * * * * * * * *
00AF D3        221 TXRETURN SEP   R3              RETURN TO CALLER
00B0 BE        222 TXENTRY  PHI   R14             SAVE CHAR IN D
00B1 E2        223          SEX   R2              RESET X REG TO R2
00B2 52        224          STR   R2              SAVE ON STACK ALSO
00B3 8B        225          GLO   R11             GET CURRENT CHECKSUM
00B4 F4        226          ADD                   PERFORM ADD
00B5 AB        227          PLO   R11             AND SAVE AWAY
00B6 F800      228          LDI   0               D=0
00B8 AA        229          PLO   R10             CLEAR PARITY COUNT
00B9 F808      230          LDI   8               8 BITS PER CHARACTER
00BB AE        231          PLO   R14             NUMBER OF BITS TO GO
00BC 7A        232          REQ                   SET START BIT
00BD DF        233          SEP   R15             DELAY ONE BIT TIME
00BE 9E        234 TXSHIFT  GHI   R14             GET BITS TILL NOW
00BF 76        235          SHRC                  GET HIGH BIT INTO DF
00C0 BE        236          PHI   R14             SAVE SHIFTED VALUE
00C1 C7        237          LSNF                  SKIP IF NO BIT
00C2 1A        238          INC   R10             INCR BIT COUNT
00C3 CF        239          LSDF                  TEST IF BIT WAS THERE
00C4 7A        240          REQ                   NO...SO SET SPACE
00C5 38        241          SKP                   SKIP THE SEQ
00C6 7B        242          SEQ                   SET TO MARK
00C7 DF        243          SEP   R15             DELAY ONE BIT TIME
00C8 2E        244          DEC   R14             NUMBER OF BITS TO GO
00C9 8E        245          GLO   R14             INTO D
00CA 3ABE      246          BNZ   TXSHIFT         IF NOT DONE YET
00CC 8A        247          GLO   R10             GET PARITY BIT COUNT
00CD 76        248          SHRC                  CHECK IF ODD # OF BITS
00CE CF        249          LSDF                  SKIP OF ODD # OF BITS
00CF 7A        250          REQ                   EVEN-SET PARITY BIT=0
00D0 38        251          SKP                   HOP OVER NEXT INSTR.
00D1 7B        252          SEQ                   ODD-SET PARITY BIT=1
00D2 DF        253          SEP   R15             DELAY FOR PARITY BIT
00D3 7B        254          SEQ                   SET STOP BIT (MARK)
00D4 DF        255          SEP   R15             DELAY ONE BIT TIME
00D5 DF        256          SEP   R15             ONE MORE...
00D6 30AF      257          BR    TXRETURN        RETURN TO CALLER
               258 *
               259 *         HEX KEYBOARD INPUT ROUTINE
               260 *
00D8 D3        261 KEYRTN   SEP   R3              RETURN TO CALLER
00D9 3FD9      262 KEYENTRY BN4   *               WAIT FOR ENTRY
```

```
LOCN OBJ CODE  STMT    SOURCE STATEMENT                    1802 VER 1.6

00DB 37DB       263        B4    *            DEBOUNCE A BIT
00DD 6C         264        INP4               GET DATA
00DE 64         265        OUT4               DISPLAY DATA
00DF 22         266        DEC   R2           FIX STACK UP
00E0 30D8       267        BR    KEYRTN       RETURN NOW
00E2            268        END

   0 DIAGNOSTICS GENERATED
  21 SYMBOLS

SYMBOL TABLE:
CHECKREG  0009    STACK     00D0    LOADSTRT  0000    HEADER    000F
LOADLOOP  001E    RXRETURN  0032    RXENTRY   0033    RXSTART   0036
RXLOOP    0042    RXBITOFF  004A    NOPARITY  0055    DELRTN    0065
DELAY     0066    DELDECR   0069    DUMPSTRT  006F    WRITDATA  009D
TXRETURN  00AF    TXENTRY   00B0    TXSHIFT   00BE    KEYRTN    00D8
KEYENTRY  00D9
```

"IPSO FACTO" STANDARD (CONT'D)

MORE SOPHISTICATION ALSO REQUIRED

Another, more sophisticated standard is also required to support things such as named files, block counts, a seperate checksum for the header, etc.

Refer to IPSO FACTO Issue #3, page 36. If you have any ideas, please write them up and submit them to any of the club Executive.

REFERENCES:

1. IPSO FACTO #3, P. 36
2. IPSO FACTO #6, P. 5
3. BYTE's Audio Cassette Standards Symposium, Byte Magazine, February 1976, p. 72.

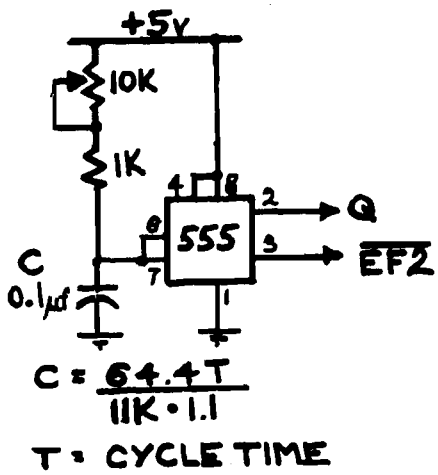
A/D NOTES

Russ Nelson
Box 1191, CCT
Potsdam, N.Y. 13676

I'm submitting a funny little circuit that works, and another that might work.

An A/D converter that converts the position of a pot to a number between 0 and 3FH.

HARDWARE

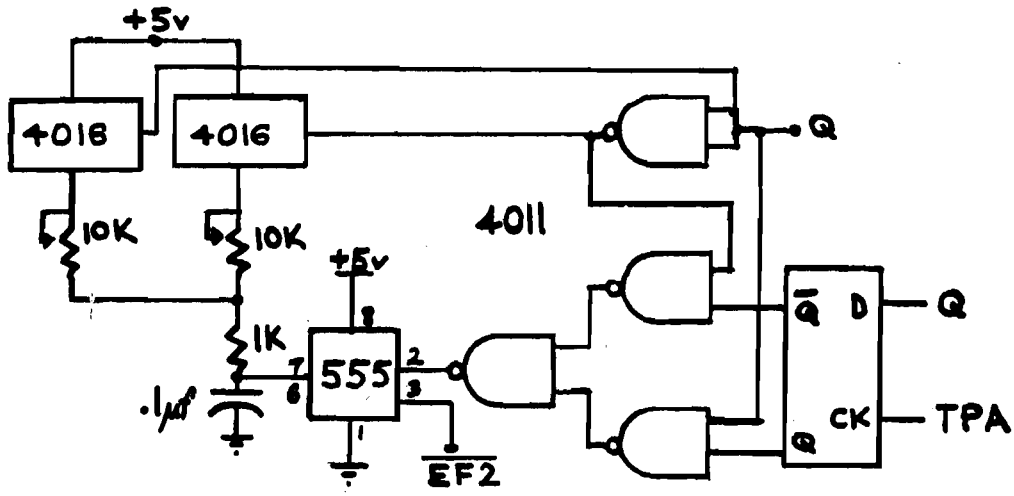


SOFTWARE

```
REQ
SEQ
LDI  0
ADI  1
BNZ  *-1
SMI  X
```

X is chosen to make the smallest resistance give a count of zero.

This circuit should convert two pot positions to numbers. I haven't breadboarded it.



SOFTWARE

```
;Q is assumed reset
     SEQ           ; do it twice so that
     SEQ           ; X and Y have same timing
LOOP LDI 0         ; time it
     ADI 1
     BNZ *-1
     BNQ OTHR      ; X or Y?
     PLO RA        ; X -store it
     REQ           ; start timer
     BR  LOOP
OTHR PHI RB        ; RB contains X and Y
```

CHRISTMAS COMPUTER MUSIC

Chris Airhart
364 Church St.
Stratford, Ontario N5A 2S1

The following tunes were run on a TEC-1802. Please note (sorry for the pun) the comments beside the song title; these are speed modifiers, and the appropriate program modifications can be found in Issue #5, IPSO FACTO, p. 37. The modifications were for a 1 MHZ clock, so if you have a much higher clock, some songs will not sound correct.

JINGLE BELLS

```
AB AB CB AB AB CB AB AD 39 9A EB 20
AC AC BC 9C AC AB AB 9B 9B AB AA AA AB CA CD
AB AB CB AB AB CB AB AD 39 9A EB 20
AC AC BC 9C AC AB AB 9B 9B AD AD AC AA 69
```

GOOD KING WENCESLAS

```
29 29 29 AA 29 29 4D 2E 2D 2E 27 49 49 29 29 29
AA 29 29 4D 2E 2D 2E 27 49 49 AD AC AB AA AB AA
49 2E 2D 2E 27 49 49 2D 2D 2E 27 29 29 CA AD AC
AB AA 49 CC 69 20
```

WHAT CHILD IS THIS? (3/2 Duration)

```
2E 49 AA BB 95 AB CA 27 3D 1E 27 49 2E 3E 16 2E
47 26 4B 2E 49 AA BB 95 AB CA 27 3D 1E 27 39 17
2E 36 15 26 6E 6E ED BD 95 AB CA 27 3D 1E 27 49
2E 3E 16 2E 47 26 6B ED BD 95 AB CA 27 3D 1E 27
39 17 2E 36 15 26 6E 6E 20
```

WE THREE KINGS (3/2 Duration)

```
CB AA 59 2E 27 29 27 6E CB AA 49 2E 27 29 27 6E
49 29 CA AA CB AB AD AC AB AA AB AA 49 27 6E 67
CA 49 29 49 2D 49 2E 49 20 49 29 49 2D 49 2E 69
49 29 CA AB CC AB CA AB 49 29 49 2D 49 2E 69 20
```

JOY TO THE WORLD

```
CD B5 9B EA 29 47 4E 6D AA EB AB E5 A5 ED AD AD
A5 AB AA BA 19 27 AD AD A5 AB AA BA 19 27 27 27
27 27 17 19 EA 19 17 2E 2E 2E 1E 17 69 17 1E 2D
CD AB BA 19 27 29 47 4E 6D 20
```

THE FIRST NOEL (3/2 Duration)

```
17 1E 3D 1E 17 19 CA 9B 95 AD A5 AB CA 9B 95 AD
A5 AB AA AB A5 AD AA 29 47 17 1E 3D 1E 17 19 CA
9B 95 AD A5 AB CA 9B 95 AD A5 AB AA AB A5 AD AA
29 47 17 1E 3D 1E 17 19 CA 9D 95 CB 9D 9E C7 AE
AD A5 AB AA AB A5 AD AA 29 47 20
```

IT CAME UPON A MIDNIGHT CLEAR (3/2 Duration)

```
2E C5 AB AB AA 27 4E 27 4E 2E 27 28 AA AA AB A5
EB 20 A5 C5 28 AB AA 27 4E 27 4E 2E 47 27 28 27
2E EA 20 A5 C5 25 25 26 2F 47 28 CA A5 AB AA 28
27 28 27 6E 20 2E C5 28 AB AA 27 4E 27 4E 2E 47
27 28 27 2E EA 20
```

GOD REST YE MERRY GENTLEMEN (3/2 Duration)

```
2E 2E AB AB AA 29 27 2E 2D 2E 27 29 AA CB 2E AB
AB AA 29 27 2E 2D 2E 27 29 AA CB 20 AB AC AA AB
AC AD AE AB AA 29 2E 27 29 CA 29 AA CB AC AB AB
AA 29 27 4E 19 17 2E CA 29 AA AB AC AD AE AB AA
29 27 6E 20
```

HARK! THE HERALD ANGELS SING

```
4C 4F 6F 2E 4F CA CA 49 CC CC EC A4 CA 49 EA 20
4C 4F 6F 2E 4F CA CA 49 CC 49 69 2E 4E 4D 4C 20
CC CC CC 4F C4 CA CA 49 CC CC CC 4F C4 CA CA 49
ED AD CD CC C4 CA E4 49 AA A4 EC 2F 4F 49 EA 20
ED AD CD CC C4 CA E4 49 AA A4 EC 2F 4F 49 6F 20
```

WHILE SHEPHERDS WATCHED

```
4F EA AA 49 4F C4 C4 CA 49 CA CC CC CB EC 20 CA
ED AC C4 CA 49 4F 4E CA 49 4F 4F 4E 6F 20
```

O COME ALL YE FAITHFUL (3/2 Duration)

```
28 48 26 28 C4 46 AC A4 AC A5 CC C4 48 29 2F 29
28 A4 AC 49 3F 16 66 20 C6 A5 AC C5 CC A4 AC 28
A4 49 26 28 28 29 28 A4 48 26 AC AC A4 AC A5 CC
A4 AC A5 AC A4 28 49 28 A5 CC B4 18 68 20
```

AWAY IN A MANAGER

```
4C 4F 4F 29 AA 4F 4F AA A4 CC CC CD E4 20 29 AA
C4 C4 CC CA CA 2F AA 49 4D 4F 6E 20 4C 4F 4F 29
AA 4F 4F AA A4 CC CC CD E4 20 29 AA C4 C4 CC CA
CA 2F AA 49 4D 4E 6F 20
```

DECK THE HALLS (1/2 Duration)

```
EC A4 CA 49 4F 49 CA 4F 29 AA A4 29 EA 29 4F 4E
6F 20 EC A4 CA 49 4F 49 CA 4F 29 AA A4 29 EA 29
4F 4E 6F 20 69 AA C4 49 EA A4 CC 49 AA AB CC AD
AE CF CE CD EC 20 EC A4 CA 49 4F 49 CA 4F AD AD
AD AD EC A4 CA 49 6F 20
```

O LITTLE TOWN OF BETHLEHAM

```
CB CE CB C4 CB CD CC 6E CA 49 27 29 CA 4D
EB 20 CB CB CB CE CD CD CC 4E CA 49 27 29 CB CA
69 20 CB CB CB CA 49 67 27 47 47 4E 47 49 CA
EB 20 CB CB CB C4 CB CD CC 4E CE CD 49 EB AA 69
```

SILENT NIGHT (Double Duration)

```
39 9A 29 6E 39 9A 29 6E CD AD EB CC AC 69 CA AA
BC 9A AA 39 9A 29 6E CA AA BC 9B AA 39 9A 29 6E
CD AD AF 9D AB EC CE 20 BC 19 2E 39 1F 2D 6C
```

O CHRISTMAS TREE

```
2D 39 19 49 AA BB 9B CB AB AA AB CC 49 CA 49 20
2D 39 19 49 AA BB 9B CB AB AA AB CC 49 CA 49 20
AD AD AB CE AD AD AC CC AC AC AA CD AC AC AB CB
4D 39 19 49 AA BB 9B CB AB AA AB CC 47 CA 49 20
```

ANGELS WE HAVE HEARD ON HIGH

```
AB AB AB AD BD 9C CB AB AA AB AD BB 9A 49 AB AB
AB AD BD 9C CB AB AA AB AD BB 9A 49 CD 9E 9D 9C
9B CC 9D 9C 9B 9A CB 9C 9B 9A 19 BA 1D 4D 29 AA
AB AC CB CA CD 9E 9D 9C 9B CC 9D 9C 9B 9A CB 9C
9B 9A 19 BA 1D 4D 29 AA AB AC CB CA 69 20
```

JOLLY OLD SAINT NICHOLAS

```
AD AD AD AD AC AC CC A4 A4 A4 A4 ED 20 29 29 29
29 2F 2F C4 AA A4 AC AD EC 20 2D AD AD AD AC AC
CC A4 A4 A4 A4 ED 20 29 29 29 29 2F 2F C4 AC A4
AC AD E4 20
```

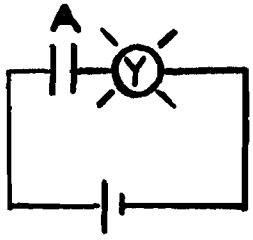
HARDWARE BASICS

Fred Feaver
105 Townsend Ave.
Burlington, Ont. L7T 1Y8

Some of our readers may be having difficulty in understanding the function of solid state components. The basis of most solid state is the various arrangements of gates. Gates are: AND, OR, NAND, NOR, NOT, EXCLUSIVE OR, AND/OR ETC.

Solid state gates can be simulated by mechanical contacts or switches as shown below:

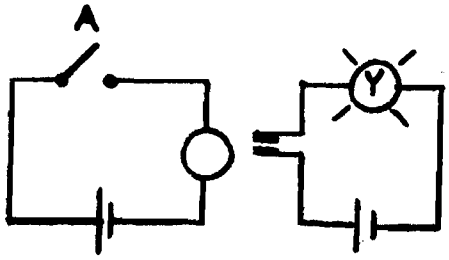
SIMPLE GATE - (Analogous to a non inverting buffer/driver)



Contacts as shown will not pass current

A = Y

When contact A is closed (contact made) Y = LIGHT or when A=1, Y=1

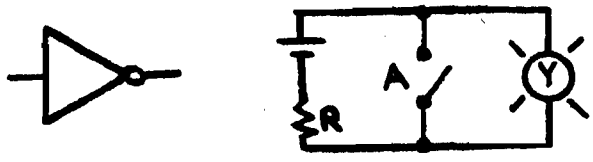


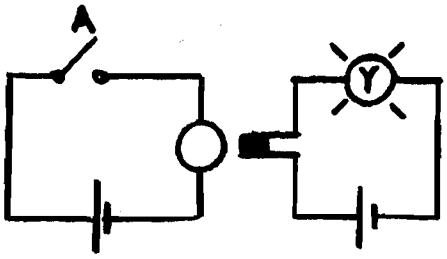
Once more A = Y

With making contacts shown, when switch A is closed, the coil is energized and the contact is closed, causing the light Y to illuminate

NOT GATE (Inverter or Inverter Buffer/Driver)

Contacts as shown pass current





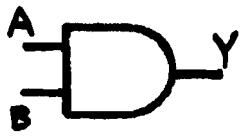
With breaking contacts:
When switch A is open (as shown left) the coil is de-energized and the contact is closed, thus lighting the LIGHT Y
When switch A is closed, the coil is energized, opening its contact to extinguish the LIGHT so A is not = Y

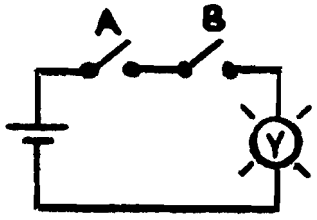
If A = 1, Y = 0
If A = 0, Y = 1 (LIT)

$\overline{A} = Y$

In alternate analogy, if switch is closed the light goes out.

"AND" GATE



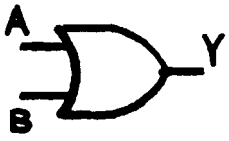


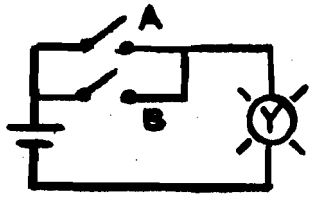
A • B = Y (A and B = Y) •Indicates AND
If A = 1 and B = 1 then Y = 1

When switch A is closed (1) and switch B is closed (1) then the light Y is illuminated (Y=1). This is the simple series circuit. There can be from 2 to 8 inputs in one IC package. All inputs must be 1 for Y to be 1.

"OR" GATE

Also called "INCLUSIVE" or
A + B = Y (A or B = Y) + indicates or
If A = 1 or B = 1 then Y = 1
or both A and B = 1 then Y = 1

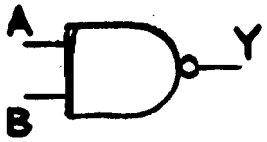


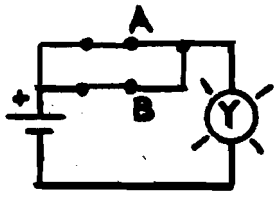


When switch A is closed (1) or switch B is closed (1) or both A and B are closed (1), then the light comes on (Y=1). This is the simple parallel (or shunt) circuit, with two or more gates in parallel.

NAND GATE

An AND gate together with a NOT gate makes a NAND Gate (NOT AND)



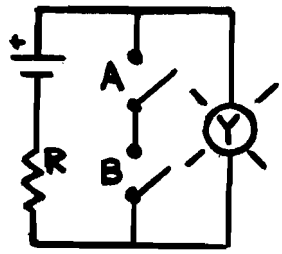


$\overline{A \cdot B} = Y$ The bar indicates negation when a switch is in its un-operated position it is considered = 0

The switches shown are normally closed

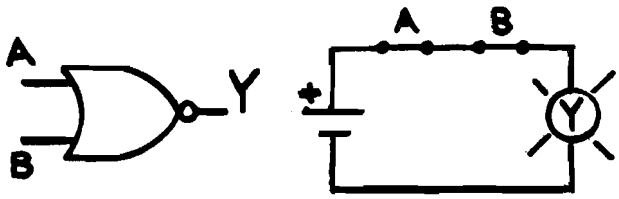
If switch A and switch B are operated (opened) (1) then the light will be extinguished (0)
So $A \cdot B = \overline{Y}$ or $\overline{A \cdot B} = Y$

ALTERNATIVE ANALOGY NAND



If A and B are closed the light goes out. $A \cdot B = \overline{Y}$ or $\overline{A \cdot B} = Y$
This is a series/shunt switching circuit.

NOR GATE

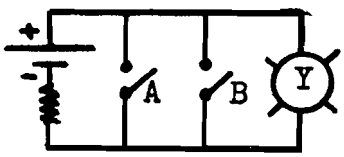


An OR GATE together with a NOT GATE makes a NOR GATE (NOT OR)

$\overline{A + B} = Y$

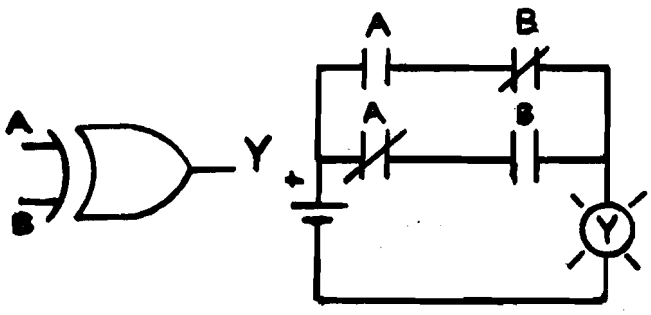
If switch A or switch B (or both) is operated (1) then the light will be extinguished (0)
So $A + B = \overline{Y}$ or $\overline{A + B} = Y$ (If neither A or B is operated the light is LIT = 1)

ALTERNATE ANALOGY NOR



If A or B is closed (or both) the light goes out
$A + B = \overline{Y}$ or $\overline{A + B} = Y$
This is a parallel switching circuit

EXCLUSIVE OR GATE



$A \cdot \overline{B} = Y$
$\overline{A} \cdot B = Y$
or $A \oplus B = Y$

Either gate, but not both will give an output

If contact A is operated the light goes on
If contact B is operated the light goes on
But if both A and B are operated then the light is out

$A \cdot \overline{B} + \overline{A} \cdot B = Y$

This can be used to generate a parity bit.

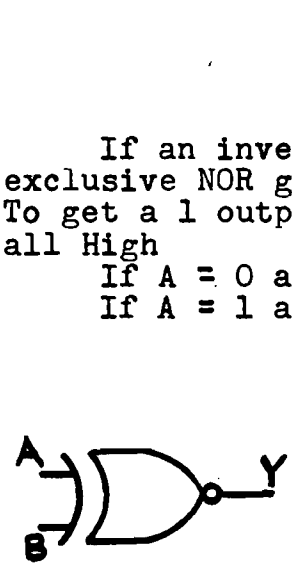
EXCLUSIVE NOR GATE (COMPARATOR)

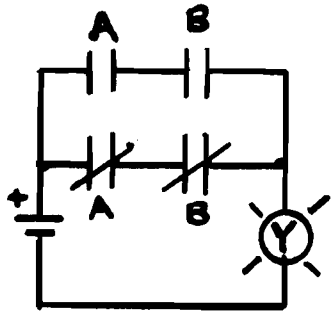
If both A and B, operated together, have the same value then Y = 1 see contact sketch

If an inverter is added to the exclusive OR gate we get an exclusive NOR gate -- used for comparing bits for equal magnitude. To get a 1 output all inputs must be the same:-either all Low or all High

If A = 0 and B = 0 then Y = 1 $A \cdot B + \overline{A} \cdot \overline{B} = Y$

If A = 1 and B = 1 then Y = 1 $A \odot B = Y$

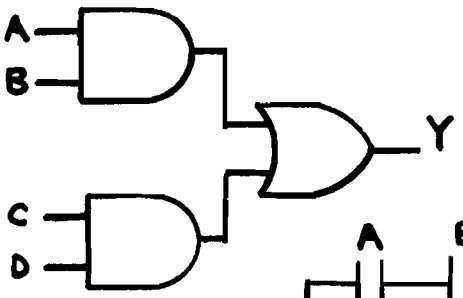




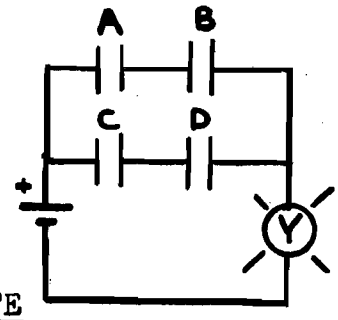
$AB + \overline{AB} = Y$

$A \odot B = Y$

AND/OR GATE



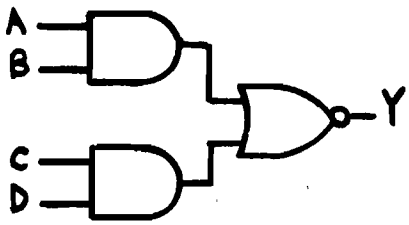
$AB + CD = Y$



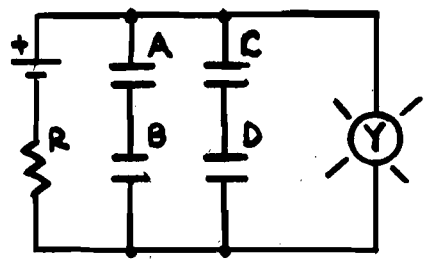
If A and B are closed Y = 1
or if C and D are closed Y = 1

AND/NOR GATE

$A \cdot B + C \cdot D = \overline{Y}$



If A and B are closed Y = 0
or if C and D are closed Y = 0



GAME OF LIFE UPDATE

Ben H. Hutchinson, Jr.
30 Partridge Rd.
Lexington, MA 02173

It's good to see LIFE in print. I already have two letters from people who got it running; one mentioned Tekatch using it in a demo at the Micro Age '78 in Toronto, June 1-2.

I was impressed in the way LIFE was retyped and assembled; so much so that I'll overlook all my "neighbors" becoming "neighbours"! I found only one trivial typo: in the "PI" pattern, the second address should be 067B, not 0678 as printed.

I have built a 64 X 128 display buffer. I am afraid it will be a while before I get it documented in publishable form. However, it works, and looks nice with LIFE and the VIP cursor drawing game. I rewrote the VIP CHIP-8 interpreter for the expanded resolution display. I call the new interpreter CHIP-10, since it controls 10 bits worth of display memory rather than 8. It works fairly well, but still has a few bugs.

I would like to hear from anyone who gets the LIFE program running. I've enclosed some new forms we've discovered, and also a CHIP-8 program written by Udo C. Pernisz of Newark, Delaware which allows drawing figures using 8-way cursor control on a VIP.

PERNISZ' 8-WAY DRAWING CURSOR WITH LINK TO HUTCHINSON'S GAME OF LIFE (CHIP-8)

Use keys 1, 3, 7, 9 for diagonal. 5 turns cursor ON. 0 turns cursor OFF.

```
0200   A258     I = 0258
  02   F865     V0:V8 = MI
  04   024A     DO MLS @ 024A
  06   00E0     ERASE
0208   F615     SET TIMER = V6
  0A   FA07     VA ← TIME
  0C   3A00     SKIP IF VA = 00
  0E   120A     GO 020A
0210   A26B     I = 026B      ("SPOT" BYTE)
  12   D121     WRITE @ V1, V2
  14   3F00     SKIP IF NOT ALREADY 1
  16   D121     WRITE AGAIN SINCE 0212 ERASED
0218   E8A1     SKIP IF KEY ≠ V8 = 0E
  1A   0306     GO TO "LIFE" AT 0306
  1C   E4A1     SKIP IF KEY ≠ V4 = 00
  1E   8740     V7 = V4 = 00
0220   E5A1     SKIP IF KEY ≠ V5 = 05
  22   8750     V7 = V5 = 05
  24   4700     SKIP IF V7 ≠ 00
  26   2250     DO  0250
0228   A262     I = 0262
022A   F065     V0 ← MI
  2C   4000     SKIP IF V0 ≠ 00
  2E   1208     GO 0208 (KEY SCAN FINISHED)
0230   E09E     SKIP IF KEY = V0
```

GAME OF LIFE UPDATE (CONT'D)

```
0232  122A    GO 022A
  34  69FF    V9 = FF
  36  70FF    V0 = V0 + FF
0238  8035    V0 = V0 - 03
  3A  3F01    SKIP IF VF = 01
  3C  1242    GO 0242
023E  7901    V9 = V9 + 01
0240  1238    GO 0238
  42  7002    V0 = V0 + 02
  44  8104    V1 = V1 + V0
  46  8294    V2 = V2 + V9
  48  1208    GO 0208
024A  01F8
  4C  058B    MLS; DRAW PICTURE ON PAGE 5
024E  E2D4
0250  D121    WRITE SPOT
  52  4F00    SKIP IF VF ≠ 00
  54  D121    WRITE SPOT
0256  00EE    RETURN

0258  0020    DUMMY VAR. V0          INITIAL X, V1
  5A  1003    INITIALY, V2           SUBTRAHEND VAR., V3
  5C  0005    "OFF" KEY, V4          "ON" KEY, V5
  5E  0C00    TIME DELAY V6          CURSOR "OFF" TO START, V7
0260  0E00    "EVOLVE" KEY, V8       (NOT USED)
  62  0102
  64  0304    KEYS SCANNED SEQUENTIALLY
  66  0607
0268  0809
  6A  0080    CLR V0, EXIT LOOP      "SPOT" BYTE
```

-Load Hutchinson's Machine-Language "Life" (VIP. Version, AS LISTED) from tape into PAGES 3 and 4. THEN, make the following changes:

ADDRESS	NEW BYTE	WHY?
030D	04	SET BACK POINTER (R2) & SUB. COUNTERS TO P.4
0327	C8	INHIBIT ERASE (0460-0479 NOT USED NOW)
0328	04	PROTECT PAGE 1, JUST IN CASE!
031F	05	USE PAGE 5 (AND 7) FOR DISPLAY (CHIP-8 NEEDS PAGE 6)

- DRAW LIFE FORMS THEN, PRESS KEY E TO "EVOLVE". THAT'S ALL!!

LIFE FORMS NOT PUBLISHED IN SCIENTIFIC AMERICAN

The accompanying pages show some interesting life forms discovered by my son Ed and I. They are most easily entered using the cursor drawing option; however, if you don't have CHIP-8, it is easy to code them up by hand. They are drawn on grid-square paper to make this easier.

I have not tried to duplicate or summarize the wealth of material published in the Mathematical Games column of Scientific American between October, 1970 and February, 1972. Anyone seriously

interested should start by going to the nearest large public or college library and looking up this material.

My original program listing and write-up of LIFE gave four examples of interesting starting states taken from Scientific American, and also an example on how to code the patterns for entry into memory. Two of these (R Peatomine and Pi) are examples of simple patterns with very long and complicated lives. The pulsar is an example of a periodic or oscillating form, while the glider is an example of a moving form. The types are listed in order of increasing rarity; very few moving forms are known and we have not discovered any, though we have found many simple long-lived forms and a few periodic oscillators.

I am eager to hear from anyone who discovers new LIFE forms, or who has any published material not printed in Scientific American, such as the LIFELINE newsletter apparently published 1971-73 by Robert Wainwright. Write me at 30 Partrigde Road, Lexington, Massachusetts, 02173.

On a 32x64 display, big patterns hit the edges, which distorts the growth. It is important to position the starting forms to postpone this as long as possible; most patterns do not grow symmetrically. Thus, patterns will sometimes seem to have different histories and lifetimes, depending on where they are started. I recently modified both hardware and software to provide 64 x 128 cells on the display; this helps. (Diagrams 55-60)

ACE TUTORIAL

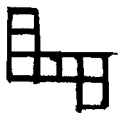
To date there have been three tutorials at the Wilcox St. location. These tutorials are aimed to provide the beginner with basic knowledge of 1802 programming and to investigate use of the device by designing and building small systems or controllers for a variety of functions. Currently the group is designing circuits/software for a motor speed control based on a temperature sensor.

Anyone wishing to attend please feel free to do so.

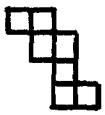
Wilcox St. Nov. 14, 7:00-8:00
Nov. 28, 7:30
Dec. 12, 7:00.

Rod Dore
660 Oxford Rd.
Unit 32
Burlington, Ont.
L7N 3M1

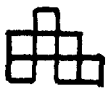
THE LONG-LIVED HEPTOMINOES



MORE THAN 200 MOVES.
START ABOVE CENTER

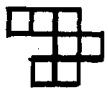


100+ MOVES. HITS TOP ON 32X64.
START BELOW CENTER.

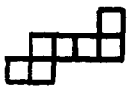


200+ MOVES. GLIDER ON MOVE 41.
START UP AND TO RIGHT OF CENTER.

STOPS AFTER 128 MOVES.
GLIDERS ON 21 & 75. START UP AND LEFT OF CENTER.

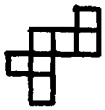


300+ MOVES. START ~5 OR 10 LEFT OF CENTER.
GLIDER ON MOVE 83.

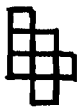


VERY LONG(~300). TURNS SOON INTO

MANY.

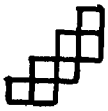


150+ MOVES. GLIDERS ON MOVES 59, 113, 140.
START UP AND RIGHT OF CENTER.

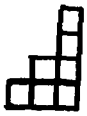


VERY LONG. MAKES BLOCK AND " J "

POND IN 76. NEVER LARGE.

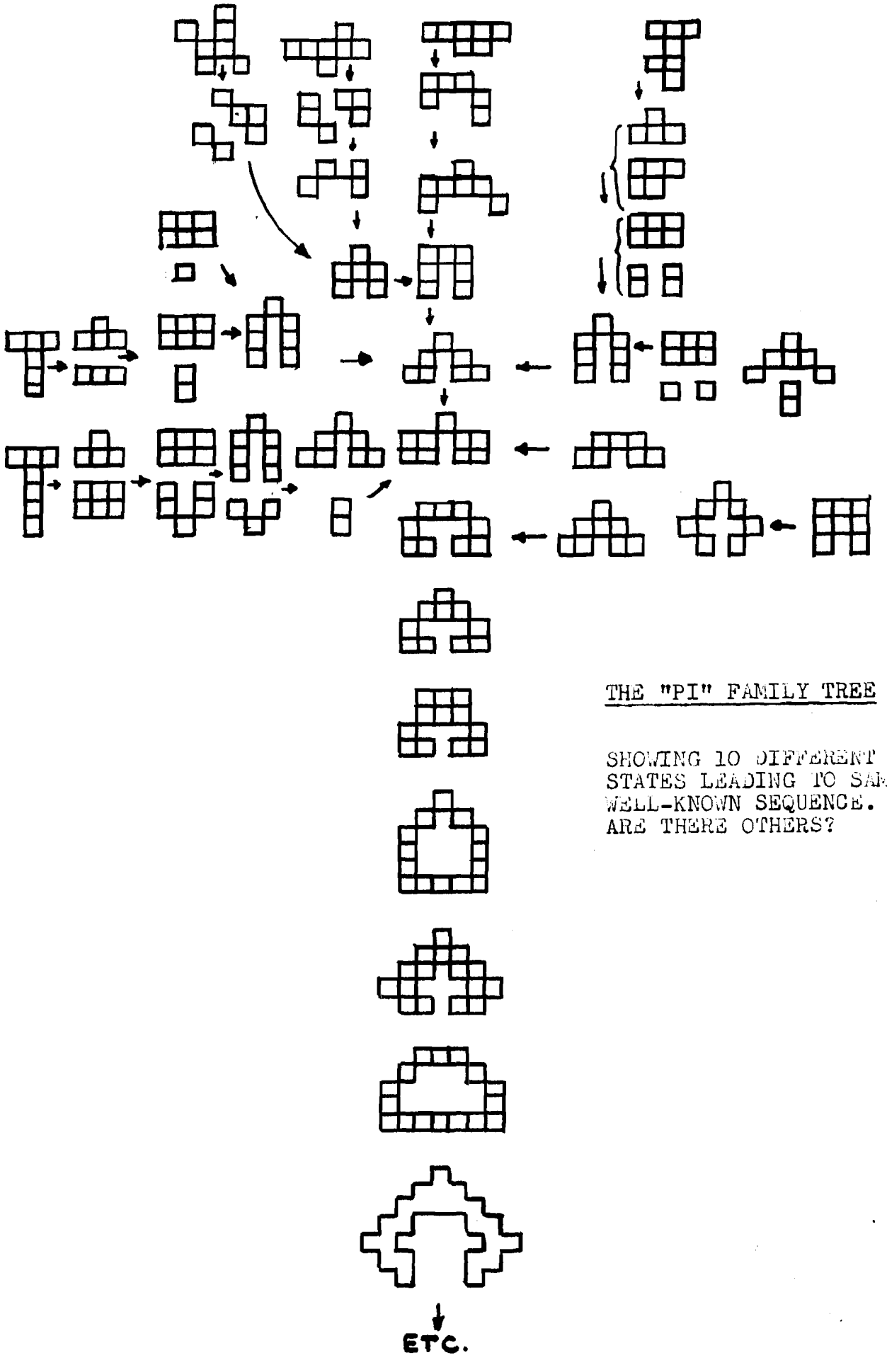


MORE THAN 200. HITS SIDES ON 32X64.



3 BLOCKS, 1 BLINKER IN ~100 MOVES.

PI! (WELL KNOWN FROM SCI. AM.)(SEE PAGE OF PI FAMILY TREE.
STABLE AFTER 173 MOVES.

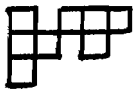


THE "PI" FAMILY TREE

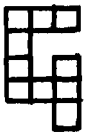
SHOWING 10 DIFFERENT STARTING STATES LEADING TO SAME WELL-KNOWN SEQUENCE. ARE THERE OTHERS?

INTERESTING LIFE PATTERNS

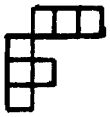
DISCOVERED BY ED HUTCHINSON
(EXCEPT (B), BEN HUTCHINSON)



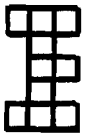
159 MOVES.
MAKES 3 BEEHIVES, 1 GLIDER



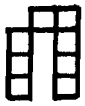
PRODUCES "PI" AT RIGHT AFTER 75 MOVES, (PLUS OTHER STUFF)
WHICH GOES INDEPENDENTLY FOR 20 OR 25 MOVES.
>>100 MOVES. HITS TOP.



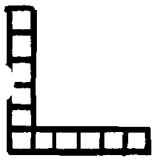
STAYS SMALL. LOAF AND BLOCK AFTER ~90 MOVES.



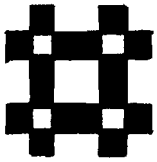
MAKES 2 "PI"'S BACK TO BACK. NICE SYMMETRY.
DIES AFTER 60 OR 70 MOVES.



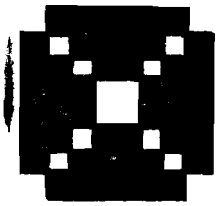
1 BEEHIVE AFTER
76 MOVES



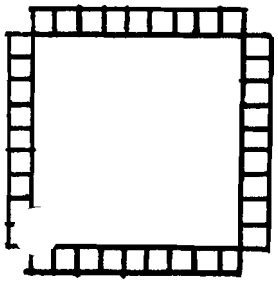
2 BLOCKS & 2 TUBS
AFTER 61 MOVES (B)
(6X6;ALSO TRY 7X7,
42 MOVES & 10X10)



OSCILLATOR-PERIOD 5

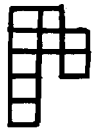


VERY NICE.
MAKES 4 PONDS.

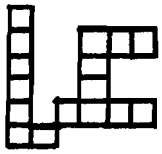


4 "PI"'S 90° APART.
- GOES ON AND ON.
(B)
(PRODUCED FROM
11X11 CHECKERBOARD
IN 1 MOVE)

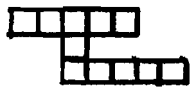
- SHORTER, BUT INTERESTING, WITH 7-UNIT SIDES.



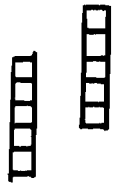
VERY LONG LIVED.
HITS ALL SIDES OF
32X64.



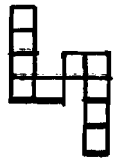
VERY LONG LIVED.
BECOMES SYMMETRICAL.
PRODUCES FOUR SQUARES.



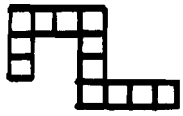
STABLE AFTER 214
GENERATIONS: 4 BLINKERS,
1 BLOCK. NEVER LARGE (B)



128 MOVES.
2 GLIDERS AFTER 63.



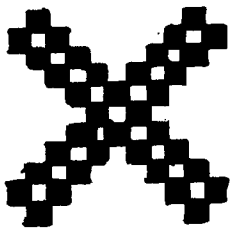
2 TRAFFIC LIGHTS
AFTER 90 MOVES.



ONE BOAT AFTER 73 MOVES.

DIES IN 46 MOVES, BUT
BEAUTIFUL. (B)

These are interesting: (make them symmetrical!)



ILLUSTRATED: DIAG. = 10

traffic lights
in 2 or 3 moves

(diagonals = 4)

diag. =6: 4 ponds in 10

diag. =8: Stable after 42 moves: traffic lights in ring of blocks

diag. =10: Dies in 22. Fairly interesting.

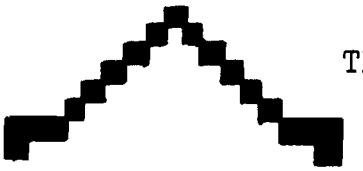
diag. =12: Remarkable stable pattern after 28 moves, nice.

diag. =14: 4 traffic lights after 31 moves (hits top and bottom)

diag. =16: Pond after 38 moves. Hits top or bottom.

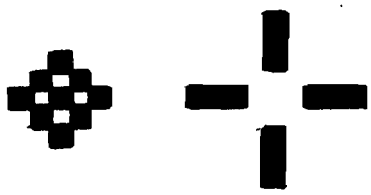
diag. =18: Many moves, 100; hits all sides of 32X64

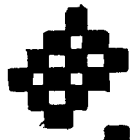
Single-row-diagonal X's and V's do interesting things. They last longer and are more interesting if the ends are "anchored" as shown so that the arms do not "burn" as "fuses".

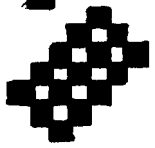


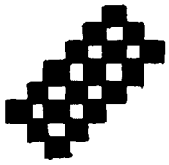
TRY DIFFERENT LENGTHS, X AND V.

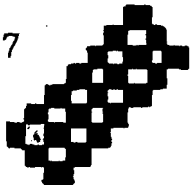
LIFE HISTORIES OF VARIOUS LENGTHS OF 3 PARALLEL DESIGNS

3  IN 5

4  DIES AFTER 31 MOVES; SPECTACULAR!

5  DIES AFTER 7

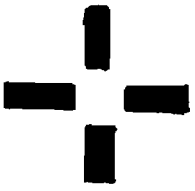
6  OSCILLATOR WITH PERIOD=8; INCLUDES

7  DIES AFTER 7

8 DIES AFTER 7

9 DIES AFTER 7

10 IN 7

11  IN 11

12 DIES AFTER 35 MOVES. SPECTACULAR!

13 DIES IN 11 MOVES

(CONT'D)

14 PERIOD-8OSC. WITH

15 DIES IN 11

16 DIES IN 11

17 DIES IN 11

18 IN 11

19 IN 13

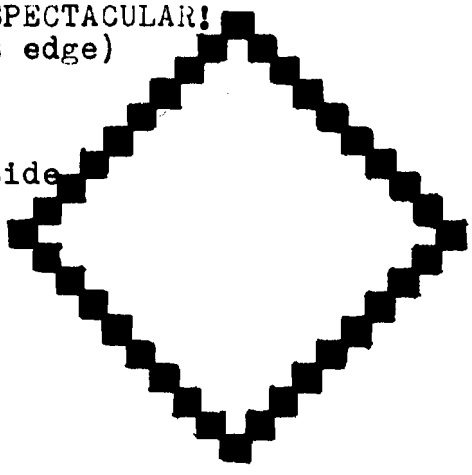
20 DIES IN 39. SPECTACULAR!

22 PERIOD-8 OSC.; AFTER 16 MOVES

27 IN 21

28 DIES IN 43. SPECTACULAR!
(one end bits edge)

A diamond, 10 one side is nice: (smaller, similar figures die quickly)



issue =8

LETTERS TO THE EDITOR

Dear Tom:

I appreciate the hard work you and your club have put forth in publishing the 1802 newsletter. It's been very helpful to me and my fellow users in Minnesota. Attached is my renewal form for it for the coming year.

I am a member of and the key contact for the 1802 in both the Minnesota Computer Society and the Scheme Computer Club. My contacts with other people in Minnesota show that the popularity of the 1802 is fast and rising in this State.

I have the 1802 Netronics version with 12K of memory, their giant board for interfacing parallel, serial, and cassette read and write. I plan on increasing the size of my system somewhat. I also have a Heath H9.

I have some new sources to contact for 1802 software:

Benchmark Computer Systems
7 Hanover Place
Hicksville, Long Island, NY 11801

Send a self-addressed, self-stamped envelope asking for the 1802 programs they have available and they will send you their current information and retain your name on their mailing list for 1802 programs. They state they will write more if the public demands more for the 1802. Let's support them!

The Hilleslands
Box 105
Parkers Prairie, MN 56361

Send $3.00 for a 25-page listing of programs.

Mr. Paul C. Moews
16B Yale Road
Renwood Apartments
Storrs, CT 06268

The price for this software package is $2.50.

Also, I am sure many 1802 users know of the package put out by Infinite, Inc. in Florida for adapting the 1802 to a S100 bus. Package cost is $9.95 prepaid.

Mr. W.J. Haberhern, Jr.
Infinite, Inc.
1924 Waverly Place
Melbourne, FL 32901

In a discussion with RCA in Sommerville, PA, I learned of some new support chips for the 1802. They consist of an 18-pin 1024 word by 4 bit SOS COS/MOS random access memory. This ram, CDP 1825, would minimize chip count of larger memories. Also available is a color generator, CDP 1862, which interfaces to the 1861 video chip to give color graphics; a new PAL/Compatible TV Interface, CDP 1864--a 40-pin chip; also a new programmable I/O Interface CDP 1851--a 40-pin chip; a new multiply-divide unit, CDP 1855, for faster math on the 1802; a programmable tone generator, 256 possible frequencies in the audio range; three different latch and decoder memory interfaces--the CDP 1866, CDP 1867, and CDP 1868. All of the above chips will be available the fourth quarter of 1978.

The new CDP 1804 microprocessor will be coming out the second quarter of 1979.

All the above chips and their specs are available in the new

Cosmac Microprocessor Product Guide #MPG 180B through your local RCA distributor.

My most recent issue of IPSO FACTO is issue #6, June 1978. If there are any more recent, I would appreciate a copy.

Again, thank you for your support of the 1802; please keep up the good work. I have shown your club newsletter to the two clubs here in the Twin Cities, and they were very impressed with its size and information. They find it hard to believe there are that many 1802 users. Most people feel that the 8080's, Z80's, 6800's, and 6502's are the only microprocessors that exist, but lately quite a number of people are taking a close second look at the 1802.

Thanks again. James E. Tate, 1415 W. Jessamine #301, St. Paul, MN 55108.

Dear Tom,

I just saw the piece I sent in on the 1K Video Ram. I must compliment you (or whoever did it) on the redrawing of the circuit diagrams. They look great! I apologize for the mess they were in when I sent them. I note that I left out the example of the driver routine for my Basic. It is enclosed. I have had Tiny Basic running for about a year on it now. Does anybody else use it? Is there a better Basic for the 1802?

I have come across something that might interest some members. A fellow at DEC has produced a cross assembler for most of the popular microprocessors. One assembler handles the 1802, 8080, Z80, 6800, and a few others. It is written in DECsystem-10 assembly language so it will only run on a DEC-10 or 20. He has offered it free to any micro-hackers that want it. I have a copy. Drop me a note if interested.

I am currently interfacing a 2708 PROM programmer that was in the March '78 BYTE to my 1802 system. I will send in a blurb on how it works when I get finished (a few weeks yet). (Please do! B.M.) Best Regards, Alan Parker, 1850 Eagle Ct. Severn, Md., 21144, USA.

Dear Mr. Murphy,

Well, I have received issue #6 and have read it twice. What can I say, it's great.

I like the 1K Video Ram circuit submitted by Alan Parker (Page 29). In his article he mentioned a scroll routine which was not included in the newsletter. Would it be possible to get a copy of it? Also I would like to correspond with Mr. Parker if you could send me his address.

Hopefully in a couple of months I'll have an article or two for you. At the moment I am trying to convert my Cosmac Elf II to the S-100 Bus. I am planning a complete conversion and have most of the signals worked out on paper. All that remains is to complete the final drawings and then start wire wrapping. Hopefully I'll be powering it up in a couple of months.

I have a Logos II 8K, 250ns memory board and two Raymond digital cassette drives to interface as soon as I have the up board running. The cassette drives came from California Digital (see enclosed ad). They are computer controlled, run at 30 I.P.S. and read/write at 500,000 Bits per second. All this and they don't cost any more than a quality cassette tape recorder.
Robert C. Taubert, 10 Michael Dr. Durham, N.C. 27704

LETTERS TO THE EDITOR (CONT'D)

ME AND MY MICRO

Volker Raab
Ramtenvej 30
DK8581 Nimtofte
Denmark

First of all, I've not got it yet. Rather accidently I stumbled over an ad, describing an affordable (less than $200) and useful (TV + cassette-audio) system, based on the RCA-1802. Thats how it all started. Ever since I've been busy gathering information and parts.

At present I have gathered 1 article, 2 pieces of 2114 RAM memory and an ASCII keyboard (oh luxury!), manuals should be on their way now.

I hope to get hold of more memory and, dare I hope, a floppy disc.

Problem is, when first I have acquired the "main frame", my budget will not allow for more than an IC or a cassette now and then for some time.

At present my "mainframe" will most likely be an ELF kit. I would prefer to start with a motherboard and a heap of IC's, but I chose not to do so due to lack of experience, tools etc. I have worked with software on all levels from big mainframes to bare minis for nearly a decade. At present while working with a maxi I believe in a sort of micro-revolution.

With regard to software, I think I would likewise prefer to do it all by myself, ending with PASCAL as described in BYTE (78-9 etc). Unfortunately however, family and friends will ask what I've got for all the money I've spent and why doesn't it do something. It would be lucky, if I could buy a cassette (for the VIP) now and then, just to show what I could make it do too.

Honestly, I want to learn more about the basic hardware and even software and do some computer fun, games, etc.

I have looked in detail at the TINY PASCAL as proposed by BYTE, but at present its more the end of the road. My first needs will be an editor and an assembler running on my system. Luckily, I know a technician who might help me now and then with a hardware problem, but I have to do most of it myself.

Gentlemen,

I think you will be interested in a company we have recently formed whose main purpose is to market software and hardware for Cosmac 1802 Computers.

Our first offering is a page relocatable extended monitor. Enclosed please find information literature and my check in the amount of $10.00 to renew my subscription to your excellent newsletter. Sincerely yours. Frank Chiechi, 37 Covington Street, Huntington Station, New York, 11746

P.S. Inquiries should be directed to: Benchmark Computer Systems, 17 Hanover Place, Hicksville, L.I., N.Y. 11801

ED. NOTE. This 'Super Monitor' is described as having extensive I/O functions, including KC read/write, UART, and tone generation in software. There are also subroutines such as Print Mem. Block, CR, LF, MR(N), R(N); ASCII-to(from)-HEX; Examine (alter) memory; get address (ASCII) from keyb'd; Standard Call and Return; KC read(write); and much more. It needs 1K plus stack, and costs $18 (paper tape) or $20 (KC tape), including user's manual and hex list.

Dear Mr. York,

Enclosed is my application and $10.00 for membership in ACE.

Also mentioned was (1) a monitor modeled after Mikbug, (2) a memory expansion board to expand from 256 bytes to 8K, (3) and 1-board evaluations (bare boards(?) available to fit ELF II Netronics connector?). Tecktron Equip. Corp. was mentioned to be supplier. How much, # pins, size, etc. for these (prototype??) boards? The Kansas City interfaces and calculation chip projects also sound very interesting (!!) as do the video displays.

Even though I can not, at present, meet you (and your group) personally, I am looking forward to being a member of ACE.

P.S. My 1802 system is an ELF II. I have the Cosmac VIP manual and am hoping to find a way to put this system, at least the ROM, on a board for use on my system. I am very much a beginner in computing but will be reading a lot, taking classes, etc. to improve my knowledge. Your club is the only one I've found so far that seems to be making an effort to place the 1802 up with the other popular systems; 8080, 6502, etc.

James J. Anderson, 29849 24th Place, South, Federal Way, WA 98003.

Dear Tom,

Just a note to let you know of a situation that I heard about from attending a local 1802 club (ELF of the Valley--San Fernando Valley, that is) yesterday. RCA has sold to Radio Shack their Studio II game unit. RS is selling it for $49.95; I just ordered one. In case you are not familiar with it, it has the following; 1802, 1861, 4 1831 CMOS roms for the interpreter and (5) games, 4 2101 rams, 4515 4 to 16 line decoder for the two 10 key keypads, a 555 for tone generation and a speaker, and a built in rf modulator. If you price out the items, I don't think one could buy them for that price. The cartridge i/f for the ROM cartridges; hence one could use their own rom or ram at that interface to run your own programs, without having to mod. the unit much, if any. I'll report on what I think of it when I get mine, and what mods are required to really utilize it.

From #6 IPSO FACTO, I got the LIFE game going for both myself and a friend with an ELF II; I wrote the keyboard routine to enable data entry to create the initial pattern, for both also. And he punched up the object code on tape so we could enter it into our systems. Unfortunately, the source listing must of been a zerox reduced copy, and it was difficult to read many of the hex char. of the object code. I finally had to use my disassembler and compare mnemonics to find the one transcription error or reading error that occurred initially. It ran, but had what appeared a 'digital fold-over' problem; appeared that patterns at the bottom were causing patterns to be generated at the top. I even wrote to Hutchinson, and he wrote that it should not do that, and what lines of code were supposed to prevent it. So that was where the disassembler was put to use; I should of suspected the reading of the listing would be a source of errors; but since the program ran fine otherwise, I thought it to be a program error. That sure is a fascinating program to watch. I have 'lost' many hours doing just that; being entranced watching the generations; some continued for 15 minutes. I wrote a keyboard routine that presently allows entry to any address on the display page, and in byte form entry of a 8 bit pattern that can be

entered horizontally or vertically. Do want to include the diagonals also. The keyboard routine in my version includes what originally followed the erase routine, that returned back to the VIP opsus. So the Life program is self contained in both mine and the ELF II version.

I am considering adding dynamic rams to at least one of my systems. Probably will be 4116's. Have a source at about $15 each; that would be $120/16K or $30/1K which could be considered cheap, and they are quite the state of the art, hobbyist wise. I know they will come down even further, but there is some point where it isn't worth the wait. My plans are to get the 16 K and get it working with the 1802. The refresh timing I have yet to look into fully, but that is about one of the earliest priorities at present.

There are several ways to do the refresh. One article I recently read indicated that a good way to do refresh is by exception; the refresh would always occur, except when the uP needed access. Therefore the pause, or DMA should not affect things. If the uP timing signal is not present in any of the expected modes of operation, an external counter could then come in to play. I feel it is unfortunate that when they designed the 1802, they didn't also allow the TP A AND B timing to always occur. I think those could have been used by the refresh control logic. However, there appears to be several signals that could be used to inhibit the refresh (MRD, MWR) so that the 'by exception' rule could be observed. If I am wrong, shoot me down.

Well I must close for now. Will remain in contact. Until Later Harley Shanko, 15025 Vanowen St., Apt. 209, Van Nuys, CA, USA 91405.

Dear George,

Enclosed please find a postal money order for $10.00 to renew my membership. The June issue of IPSO FACTO has just reached me and it was loaded with valuable ideas.

I'm currently using a Netronics ELF with their expansion board cassette I/O etc. and a homebrew 1K memory board. I'd like to get 4K of memory sometime this year and perhaps use the 1K in a video display as per your last issue. I'm also planning to build the touch display described in the last two issues of BYTE. This encoder is especially interesting to me since my wife works with physically handicapped children. A "non-contact" type of display such as this would be ideal for them.

I realize it's now early September but if it's at all possible I'd like to get a copy of the August IPSO FACTO. I've tried several different newsletters and either they never really got started or they folded up prematurely. I hope the Association of Computer Experimenters keeps going strong. Sincerely, Warren Jochem, 6F Booker Creek Apts., Chapel Hill, N.C. 27514, USA

Dear Mr. Crawford,

Being an 1802 user, IPSO FACTO is a welcome addition to my library.

I have modified portions of the RCA COSMAC VIP operating system and the CHIP-8 interpreter for use on an 'ELF' type system using the 1861 video chip and 4K of memory.

The $5 RCA VIP manual, in my opinion is a software jewel. The video game package alone lets the whole family enjoy your system.

Even neigbours and friends actually become interested in your hobby when they operate your system with 19 games to choose from.

If there is sufficient interest, I will prepare an article for IPSO FACTO describing how to modify the software. No hardware changes are really required.

Keep up the good work. Sincerely, Lynn R. Clock, 215 Pegasus Ave., Lompoc, CA., 93436.

Dear Tom:

I've gotten the 1802 Software Blues. I am sick of seeing all this Software for the 8080 or Z-80. I mean let's face it there just isn't as much or even half as much software for the 1802. I was thinking, since your the editor of a newsletter and Ken Bevis is President of a User's Club, maybe if you write letters to independent companies and make them realize that we (1802 users) are here, they may put out software and hardware and make the 1802 a leading chip in uP's. (You should also contact other Club's and have them write.) All we need is a good company to back us up.

I also have to say that each IPSO FACTO I receive gets bigger and better all the time. Keep up the good work. Thank you. Yours truly, Alberto Magnani, 22-68 28 Street, Astoria, New York, 11105

Dear Tom,

Thank you for your note and the membership application.

It will still be a few weeks before my system is up and running but once it's up I hope to be able to program it quite readily.

I can't understand why the 'IDLE' instruction uses the address BVS. It makes more sense to leave it tristated so that another CPU can reference memory. But it's a lovely chip otherwise. Yours truly, Muray Thayer, 115 Parkmount Rd., Toronto, Ontario, Canada, M4S 4V3.

Sign me up for another year. IPSO FACTO is clearly the best 1802 periodical. It would be nice to get it sooner--June issue arrived only 3 days ago.

I just sent an article off to KILOBAUD on displaying text on 1861 video (sorry--I put a lot of work into it, and they pay, tho not enough). My point is that you should probably review articles in KB, VIPER, etc. as well as DDJ. Tom Pittman, C/O Itty Bitty Computers, P.O. Box 23189, San Jose, CA., USA, 95153.

Dear Mr. Crawford:

Recently I had the pleasure of meeting Mr. Edw. M. McCormick and seeing a demonstration of his COSMAC system. During the demo, he was kind enough to show me copies of "IPSO FACTO" and he stated that your group was the most organized and advanced.

A bit of information about myself: I am employed as an electronic technician, and have a home-brew version of the Cosmac Elf, with 256 bytes of CMOS RAM and a self-designed ocillator circuit utilizing the National MM5369 60 Hz timebase generator. By using this chip, I have the divide-by-two circuit enabling me to use the Video Interface chip, and a "free" 60-Hz reference for interrupt and Real-Time-Clock applications.

For Software, I am using that which was published with the

ELF series of articles, and am blundering along writing a CW-tutor program which will operate in 256 bytes of memory.

Thank you for your service, and I hope that with the support of myself and others--that you will be able to continue to be a clearing house for COSMAC information. If I can be of any assistance, feel free to call on me. Donn Fisher, P.O. Box 5937, Huntington Beach, CA 92646, USA.

Dear Bernie,

I work for IBM in the Floppy Disk Drive area. (I was a member of ACE and inadvertantly let my membership lapse. I am presently being reinstated.) I would like to contribute but don't have much of a system yet. I am very slow--hard to find time. I am designing my ASCII keyboard interface and KC standard cassette interface. When I get these built and attached, I feel I'll have a little something to work with and will go from there to a TV interface.

Working here I have picked up a knowledge of magnetic recording and would like to share this with ACE members. Theory would be limited to words only and would try to impart a "feeling" for what's going on in the head, media, and circuitry. I would only explain saturation recording, as that is what I am most familiar with. I've never done this before and I don't know how long the article(s) would be. What I propose is to put it all in one package and let you guys split it up and edit as you see fit. I'll enclose an overview and let you decide if you want the whole thing.

I plan on building my TV interface using the Motorola MC6847 Video display generator. All you have to add is a 3.58 MHZ XTAL, 1K RAM, and it will output 5x7 dot characters ready to put into a pixie-verter. It's also capable of 256x192 graphics by increasing the RAM to 6K. I don't have a solid design yet, but there's great potential with this module. As I said, I'm veryslow. Don't expect anything soon on my TV interface.
Bob F. Herald, 1514 9th Ave. Se., Rochester, Minn. 55901.

Dear Tom,

I recently became aware of your publication IPSO FACTO based on the RCA Micro from a friend of mine who got me started with micros about a year ago. Since that time I have managed to build up a small system around the Netronics kit I started with, ASCII Keyboard 3K RAM printer, etc. My only disappointments thus far have been the kit and the lack of user information on the 1802 other than the original articles in P.E.

I would therefore, like to subscribe to your publication but could not find any mention of cost in the few copies of IPSO FACTO I have read. Sincerely, Doug Kyle, 6476 Griffiths Ave., Burnaby, B.C. Canada, V5E 2W9.

Dear Ken:

I am interested in the RCA 1802 microprocessor and now have an expanded cosmac Elf (from Quest) that is output buffered to a S-100 motherboard. I have 5K of an 8K ram implemented, with no rom as yet. If any of your old newsletters discuss how to program any Eproms I would be especially interested.

Unfortunately Ames, Ia. is a bit too far away to make any of

your meetings but I appreciate the opportunity to correspond with others who have interest in the Cosmac chip. Thank you.
Dennis DiMarco, 108 O&L Ames Lab, Ames, Ia., 50011.
P.S. Does anyone there know what happened to Charles Manrey's Cosmac Elf Newsletter? I got 2 newsletters and then nothing. I believe a subscription to IPSO FACTO is $10 so $10 is included.

ITEMS FOR SALE

1. I have a computer for sale. A Quest Electronics ELF Computer with 256 words of memory, 1802CD slow chip, toggle switches. For input, memory protect, input, load, PROM select. For output, it uses T.I. 311 Hex displays, the power supply is on the back of board with the regulator etc. On board. It has an 8 bit x 32 PROM routine as a monitor. I want the best offer above $100.00.
David Brady, Box 353, Angus, Ontario, LOM 1BO.

2. I have a wire-wrap board for sale. It is loaded with gold pin which can be removed, or left alone and used to build circuits with 14 pin ICs. The board as is, will hold 216, 14 pin ICs. Also, anyone with the right equipment and lots of ambition could modify the board to accommodate 16 pin ICs. Reply by mail to James A. Tadlock, 1519 Rockbrook, Lancaster Tx., 75146, USA. The board is 31.5 x 40.5 cm. Make an offer.

3. For sale one System 2 by the Digital Group Inc. Less than 200 hours. $2675.00. Mainframe: Z-80 CPU Board with 2K RAM and 256 Bytes PROM. Z80-CPU. I/O Board with four parallel input and output ports, present configuration is using 3 ports for audio cassette interface, keyboard input, video driver, and phideck interface. 10-F. Two 8K memory boards. MEM-'c. Phideck interface board with two decks. PHI-F. Video driver board for 32 x 16 characters and audio cassette interface. TVC-F. Power supply +5V @ 18 amp, -12V @ 1 amp, and +12V @ 1 amp. PWR-18. Cabinet with fan, power switch, reset switch, etc. CB-CPU. Cassette Storage System: 2 Computer controlled phidecks with cabinet. Data Rate: 6400 Baud. Media: standard audio cassettes, C-30s. Keyboard: 128 Character ASCII , Numeric Pad, Lighted Upper-Case Key, Capacitance Technology, 2 Key Rollover Cursor Controls. Software: Phideck Operating System (PHIMON), Assembler, Disassembler, Maxi-Basic, Text Editor. CONTACT:
C.R. Berger, 3225 Larche Crescent, Regina, Saskatchewan, S4S 1M9 (306)586-0288.

4. For sale Netronics ELF II with all edge connectors installed-with Manual. Giant Board with Manual. 4K RAM Board with Manual. All boards assembled and fully socketed. ELF and Giant Board tested. 4K RAM card not tested since power supply is not capable of supplying enough current. Wall Mount Transformer - powers ELF and Giant Board. TINY BASIC on ELF II cassette with users Manual. Tom Pittman's Short Course on Microprocessor Programming for ELF II. 1802 TINY BASIC on paper tape (from Itty Bitty Computers) with Listing, Users Manual, and Experimenters kit. RCA 1802 Users Manual. Copies

ITEMS FOR SALE (CONT'D

of the Popular Electronics contruction series for the ELF. Various articles, software source catalogues, and programs for 1802 systems. Issues 1-7 of IPSO FACTO. Total cost of system and software was over $300. WILL SELL EVERYTHING FOR $250.00. Send certified check or money order to William J. Grzanich, 400 Mill Rd., Apt 30, Addison, Il., 60101, USA. PHONE (312)543-0685

LETTERS OF CONTACT

Dear Tom,

Please let others know that several of us in central Connecticut (1 VIP, several Elfs, many on the verge) are interested in forming an 1802 users group--to be a "chapter" of ACE with IPSO FACTO as our newsletter (except for local news etc.). Use my name, address and phone to get things started. This info about Conn. users' group my be printed if you've got the room-and thanks! Steve Place, 72 Burgundy Hill Lane, Middletown, CT 06457, home: 203-632-0028, work: 203-666-1541 (ext. 261).
Who are some of the users in southern New England? Are there any users groups of which we are not aware?

I've read in the newsletter with great interest about your tutorial/training sessions. Is there any way this information might be shared? I'll personally pay for photocopies and postage for outlines/text/references/etc. We have several very interested but complete novices--male and female, young and "not-so-young". Please pass my name on to Norm Cunningham. I'm involved professionally with a binational amateur radio training program (Canada & US) with over 5000 volunteer instructors. Maybe I could lend a hand in some way.

Enough for this letter--I'm writing from the Philadelphia Sheraton--hope to see (to have seen?) some of the Ontario crew at our booth or my talks at PC'78. Best Wishes for continued success. I only wish I were more "local". Sincerely, Steve Place WB1EYI.

I would like to hear from members who have information about disk interface or Forth for the 1802. Thank you, John D. Owens, Sunnyside Campus, 715 Ocean Terrace, Staten Island, New York 10301.

Dear Bernie,

You might like to add in your newsletter that if any other member lives in this area we would welcome hearing from them as I have access to a 1802 Cosmac development system with 4K (going to 8K) and Heathkit terminal with cassette interface and some other goodies (I have the Quart system). I also hope to be sending you some new programs for your newsletter soon. Regards, Bob Ison, 1289 Balboa Ct #240, Sunnyvale, CA. 94086 USA TEL(415)965-3759.

Dear Bernie,

For anyone in Northeast Pennsylvania we are in the process of setting up a local computer club. We're not strictly an 1802 club although several members have Elf IIs and a VIP. If anyone is interested please contact me. Best Wishes, David A. Hersker, Box 802, Conyngham, PA., 18219.

LETTERS OF CONTACT (CONT'D)

I need an ASCII Keyboard that meets the following criteria. I would prefer reed switches. It must be able to decode the full 128 ASCII symbols and have all 128 symbols on the keycaps. I want a quality keyboard and will take it new, used, or kit. If anyone knows where I can get such a keyboard I would certainly appreciate hearing from them. Robert C. Taubert, 10 Michael Dr., Durham, N.C. 27704.

I would be delighted to hear from any other 1802 users. William P. Freymuth, 9119 E 66th St., Tulsa, Oklahoma 74133.

THE ASSOCIATION OF COMPUTER EXPERIMENTERS

MINUTES OF CLUB MEETING 78-7

HELD AT STELCO WILCOX ST. AUDITORIUM

12 SEPTEMBER, 1978 8:00 P.M.

78-7-1 George York, secretary-treasurer, reported 213 paid members and a bank balance of $2,411.91. The club books have not been audited yet.

78-7-2 Motion to adopt Minutes of 78-4, 78-5 and 78-6 as included in Issue #6.
Proposed - George York
Seconded - Tom Crawford
Carried unanimously

78-7-3 Deadline for the T-shirt Logo Contest is the meeting scheduled for December 12, 1978. Entries will be judged at that meeting.

78-7-4 The Training Co-ordinator, Norm Cunningham, has had to resign due to business commitments. Rod Dore and Mike Pupeza volunteered.
Motion to approve Rod Dore and Mike Pupeza as Training Co-ordinators
Proposed - Ken Bevis
Seconded - George York
Carried unanimously.

78-7-5 The members introduced themselves and mentioned their systems/aspirations.

78-7-6 Fred Feaver will co-ordinate and compile a study of hardware and software problems that are arising.

78-7-7 Tom Crawford demonstrated his system.

MINUTES OF CLUB MEETING 78-7 (CONT'D

78-7-8 Motion to adjourn meeting
Proposed - John Hanson
Seconded - Wayne Bowdish
Passed

The meeting adjourned at 10:00 P.M.
About 35 people attended.

THE ASSOCIATION OF COMPUTER EXPERIMENTERS

MINUTES OF CLUB MEETING 78-8

HELD AT STELCO WILCOX ST. AUDITORIUM

10 OCTOBER, 1978 8:00 P.M.

78-8-1 The meeting was preceded by a 1 hour tutorial.

78-8-2 George York, secretary-treasurer, reported a paid membership of 271 and current assests of $2,953.

78-8-3 Newsletter Issue #8 should be ready for the next meeting. Bernie Murphy asked for help to decrease the workload of the newsletter editing.

78-8-4 The Training Committee have had 2 tutorials. They have started with basics and tutorials are geared for the novice.

Mike Pupeza had to resign due to time conflicts--therefore Rod Dore asked for volunteers to help with the tutorials.

The comment was made that the instruction would be easier if a TVT was available.

78-8-5 Fred Feaver commented on a lack of requests for hardware help--some people mentioned their particular problems.

There were some comments on the software "cassette standard". The software is to be published in Issue #8.

78-8-6 Bert deKat discussed some applications that could be applied with the 1802 eg. thermostat type temperature sensors, moxie temperature sensors for solid state non-contacting sensors, shaft encoders, MOC3010 opto-isolator triac, CDP1855 RCA's multiply/divide chip.

78-8-7 Ken Bevis demonstrated his TVT and monitor. There was also a PET available with a blackjack program.

78-8-8 The meeting ended at 10:00 P.M.
31 people attended.

A NOTE REGARDING MEMBERSHIP DUES

G. York
60 Chester Road
Stoney Creek, Ont.
L8E 1Y2

There have been some inquiries regarding the increase of dues from $5.00 to $10.00. A few words of explanation may be in order.

1. The membership dues are meant to cover newsletter expenses eg. printing, mailing, envelopes, tax and operating expenses.

2. The issues #1 to #6 totalled 248 pages and the membership reached 465.

3. Printing expenses (not including mailing, tax, etc.) per member would have been $9.61. The total worth of the newsletter service to members last year came to $12.00 per member. We were very fortunate to obtain a lot of help from local business and industry in the form of "free" copying and supplies.

4. The help we received was invaluable in getting the club off the ground. However, this year the executive felt we should attempt to be self-sufficient--thus the dues increased to $10.00.

NOTICE OF CLUB MEMBERSHIP RENEWAL & NEW CLUB DUES ASSESSMENT

All club memberships expire 31 May, 1978. Club members should complete the form below and mail to George York, 60 Chester Road, Stoney Creek, Ontario, Canada, L8E 1Y2.

The club dues assessment for the 1978-79 fiscal year is $10.00. Make cheques payable to the Association of Computer Experimenters.

If you know of anyone who would be interested in joining our club, why not give him/her the membership application. We are constantly looking for new members with new and interesting ideas.

MEMBERSHIP APPLICATION FORM FOR THE Association of Computer Experimenter

FIRST NAME | LAST NAME

FIRST LINE OF ADDRESS

SECOND LINE OF ADDRESS

CITY & PROVINCE | POSTAL CODE

Association of Computer Experimenters

MEMBERSHIP RENEWAL FORM

ATTACH MAILING LABEL HERE

FIRST NAME | LAST NAME

FIRST LINE OF ADDRESS

SECOND LINE OF ADDRESS

CITY & PROVINCE | POSTAL CODE